ADVANCED COMPUTATIONS FOR BALLISTIC IMPACT PROBLEMS

G. R. Johnson*, S. R. Beissel, C. A. Gerlach, R. A. Stryk, A. A. Johnson and T. J. Holmquist
Network Computing Services, Inc.
Minneapolis, Minnesota, 55415

ABSTRACT

This article addresses some issues and solutions for ballistic impact computations. A discussion of the strengths and weaknesses of existing computational techniques is presented, and this is followed by a description of a new computational technique that is well-suited for ballistic impact computations. This new approach uses both finite elements and meshless particles. The initial grid is composed entirely of finite elements. Then as the solution progresses, the highly strained finite elements are automatically converted into meshless particles. Generally, most of the grid remains as finite elements, and this allows for an accurate and efficient solution for the less distorted portion of the problem. Only the highly distorted regions of the problem are converted into meshless particles, and these meshless particles can accurately and robustly represent the high distortions that the finite elements are not able to represent. Several examples are provided to illustrate this approach. Included is the capability to compute the formation of Behind Armor Debris (BAD) and to track it through large distances.

1. INTRODUCTION

There are many complexities associated with high-velocity ballistic impact. There are high pressures, high temperatures, large strains and high strain rates. A wide variety of materials can be involved. They can interact with one another and are subject to failure and fragmentation. Furthermore, these events generally occur during a small fraction of a second. Testing can be very expensive and time-consuming, and it is possible to obtain only limited data from the tests. Computations, on the other hand, can provide a detailed look into the complicated processes that occur during the course of the event. They can be performed in a parametric manner such that a wide range of designs can be considered. They can also be used to examine conditions that cannot be readily tested, such as impact velocities and materials that are not yet attainable. Ultimately, the goal is to provide the designer and researcher the computational tools required to design and analyze projectiles, armors and other systems, in an accurate and efficient manner.

The first ballistic impact computations were performed about 40 years ago. Since then there has been steady progress, but the complexities noted above have tended to make progress slow and difficult. This paper examines some of these computational issues and describes a new approach that offers significant advancements. Several examples are included to illustrate these new developments. In all cases the computations are performed with the 2003 and 2004 versions of the EPIC code (an explicit, Lagrangian code with finite elements and meshless particles).

2. TECHNICAL DISCUSSION

Some desirable characteristics of a computer code are that it is robust, accurate and efficient. This means it will always run for a wide range of applications, will give the right answers, is easy to use and will run quickly. Many of these characteristics are now becoming attainable.

2.1 Computational Approaches

Lagrangian approaches are most commonly used for structural analyses. Here the grid is embedded into the material, and it is possible to clearly track boundaries, interfaces and material histories. Finite elements are the most popular form of Lagrangian techniques, but meshless particles are being used more and more. Finite elements are limited in the amount of distortion they can accurately represent, but meshless particles can represent any degree of distortion as the particle algorithms have variable nodal connectivity. Figure 1 shows a particle node surrounded by five neighbor nodes, but these neighbors are not fixed and each particle node can acquire different neighbor nodes as the solution progresses. The neighbor nodes are used to determine the velocity gradients (strain rates) for the center node, and the stress gradients (forces between particles). Generally finite elements are more accurate and more efficient for mild distortions, but meshless particles are more robust, accurate and efficient for highly distorted material.

Eulerian approaches, where the material flows through a fixed grid, are most often used for fluid flow and response of highly distorted solids. They have been used extensively for high-velocity impact problems. Although these approaches are very robust, and can readily handle the distortions, some inaccuracies are introduced at boundaries, interfaces and material histories. Two more recent approaches are the linking together of Lagrangian and Eulerian algorithms, and the development of Arbitrary Lagrangian Eulerian (ALE) algorithms. These latter two

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
00 DEC 2004	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Advanced Computations For Ballistic Impact Problems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Network Computing Services, Inc. Minneapolis, Minnesota, 55415**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001736, Proceedings for the Army Science Conference (24th) Held on 29 November - 2 December 2005 in Orlando, Florida. , The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**36**	

approaches attempt to perform the computations of the mildly distorted regions with a Lagrangian technique and the highly distorted regions with an Eulerian technique. All three of these approaches that contain an Eulerian component are well suited for a wide range of problems.

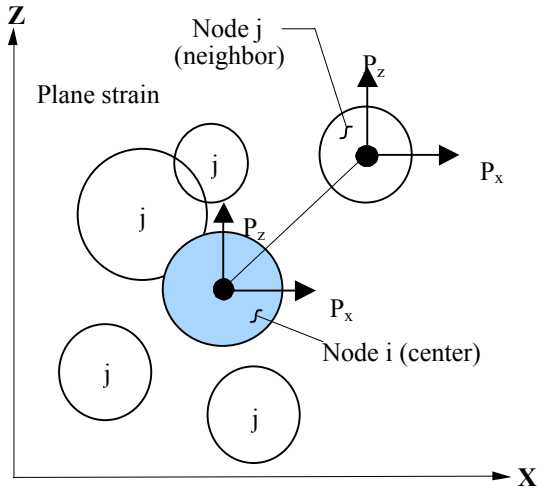


Fig. 1, Meshless particle with neighbor particles

There are some classes of problems, however, for which a Lagrangian approach is clearly preferred. These are generally problems for which it is necessary to accurately track boundaries and interfaces. Included are thin structural members, concrete structures with reinforcing steel, composite materials with high-strength fibers, and free-flight travel through large distances. A combination of finite elements and meshless particles can be used to accurately analyze these classes of problems.

2.2 Automatic Conversion Algorithm

Figure 2 shows a finite element grid with three elements on the surface (A, B, C) that are designated as candidates for conversion. An element is converted into a particle when the element has at least one side on the surface and the equivalent strain exceeds a user-specified value (in the range of 0.3 to 0.6). All of the converted element variables are transferred to the new particle node, the element is removed from the computation, and the surfaces of the remaining elements are updated. The particle is then attached to the adjacent element face until the element containing that face is converted to a particle. Details of the 2D and 3D conversion algorithms are provided by Johnson et al., 2002, and Johnson and Stryk, 2003.

There are other sliding/contact conditions, in addition to the conversion attachment algorithm noted previously, that can also occur. It is possible for the standard (finite element) nodes, and the particle nodes, to contact and slide along the external surfaces of the finite elements (Johnson and Stryk, 2001). Important components of these algorithms are the searching routines that are required to quickly and accurately identify the appropriate element faces with which the nodes can interact. When particle nodes of different materials come into contact, another contact algorithm is used. Details of this particle contact algorithm, as well as the formulation of the Generalized Particle Algorithm (GPA), are provided by Johnson, Beissel and Stryk, 2002.

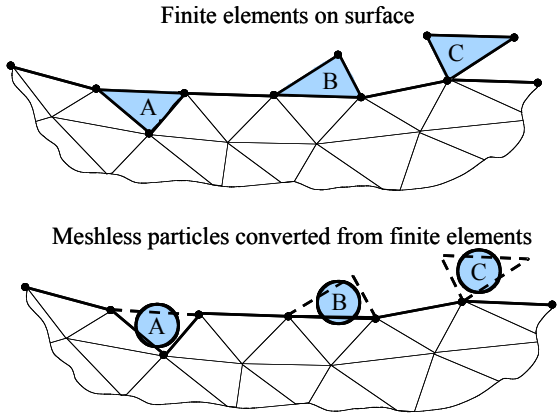


Fig. 2, Conversion of elements into particles

The finite element formulation and the computational material models are two other important components of a computer code. These will not be addressed herein, except to note that the authors have recently put a number of material models into a Modular Material Model (MMM) format. This development provides accurate, efficient and documented computational material models, and it also allows the models to be accurately transferred from one code to another with a minimum of effort. A key feature of all of these algorithms (elements, particles, sliding/contact, material models) is that they exhibit adequate robustness to handle the wide range of pressures, temperatures, strains and strain rates that occur under these ballistic impact conditions.

2.3 Efficiency

The complex numerical algorithms noted previously are of limited practical use if they cannot be used efficiently by engineers and researchers. The ability to preprocess efficiently is very important, and it has been enhanced significantly during the past decade. Looking back into the late seventies, when the first 3D computations were performed with the EPIC code, there were no existing 3D grid generators. Figure 3 is a photograph of (one quarter of) a rounded nose geometry that was generated with toothpicks and gumdrops. Each layer of elements was represented by a different color of toothpicks and gumdrops. This model was used to develop the first grid generator for projectile nose shapes in an

early version of the EPIC code. Since then many powerful grid generators have been developed, either as attached components of existing computer codes or as stand-alone grid generators that can be used with a variety of computer codes.

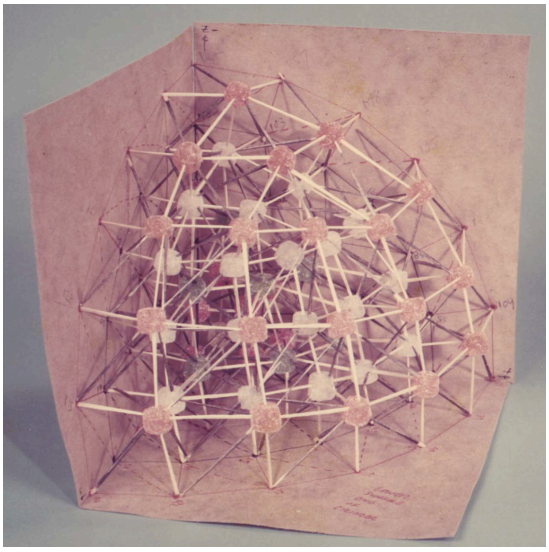


Fig. 3, An early 3D grid generation model composed of toothpicks and gumdrops

The authors have also developed a series of "Short Forms" that can be used to generate complete input files for selected problems in a matter of minutes. As an example, Figure 4 shows the Short Form input required for a 3D computation of a penetrator impacting a concrete target (with reinforcing steel) at an oblique angle. The user inputs only the dimensions of the penetrator, the size and location of rebar in the target, the materials (from a library), the impact velocity, the obliquity, and the time duration of the simulation. The grid is generated automatically, and the user has a choice of coarse, medium or fine. Complex grids and input files, for problems containing hundreds of thousands of elements, can be generated in minutes.

Postprocessing, or visualization, of results is also important and it is a challenge for large problems involving millions of elements. The Presto Visualizer is an interactive data visualization program for unstructured data sets that supports various element types and also meshless particles (A. Johnson and Quammen, 2003, and A. Johnson, 2004). It is built for remote visualization from the desktop (using a tightly coupled client-server approach), has a scalable distributed-memory parallel implementation, can visualize extremely large data sets (over a billion elements in some cases), and it is portable to almost any architecture (both desktop and large parallel systems). Its capabilities will be demonstrated later.

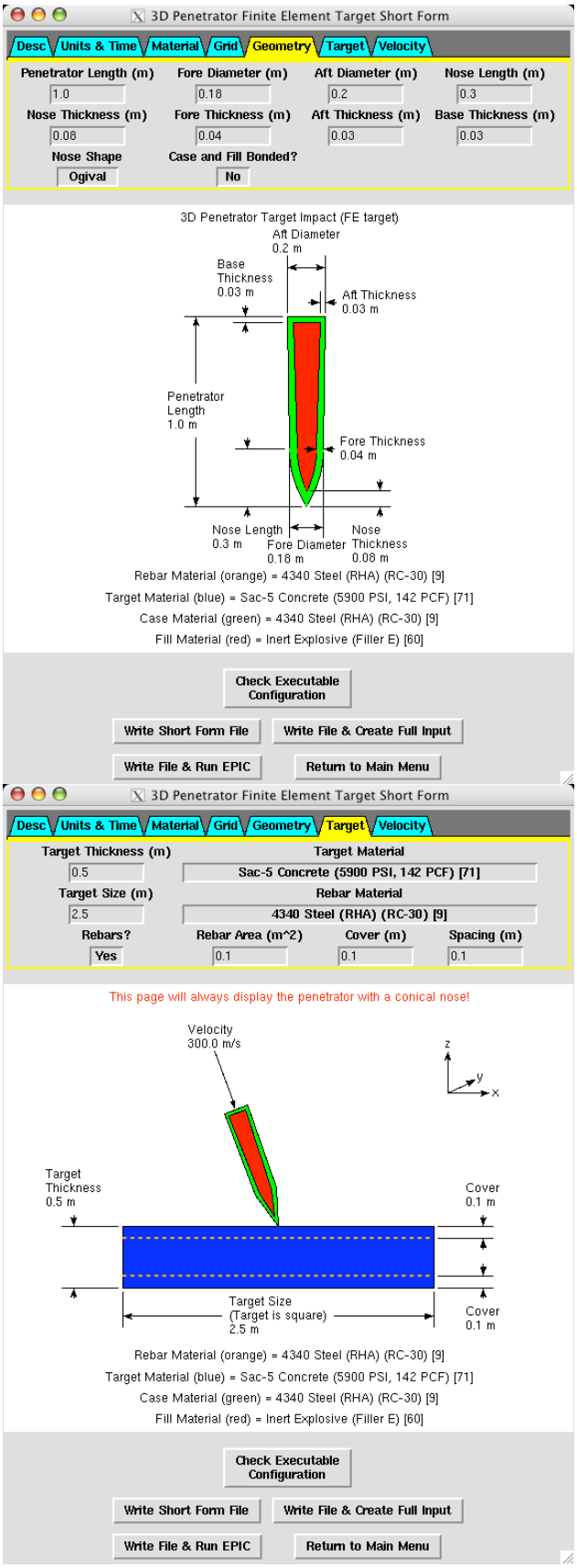


Fig. 4, Short Form for a penetrator impact problem

The final efficiency topic concerns the capability to reduce run times by effectively utilizing parallel computers. Without this capability the size of problems (numbers of elements and particles) that can be addressed is seriously limited. Although it is a straightforward procedure to parallelize finite element algorithms, the inclusion of complex sliding/contact interfaces, meshless particles, and the conversion of elements into particles, is more challenging. Parallelization of all of these features is a current effort.

3. EXAMPLES

The conversion algorithm has been verified with a number of comparisons between test data and corresponding computations. Templeton et al., 2001, presented comparisons for penetration of tungsten rods into steel targets, where the impact velocities ranged from 500 to 3000 *m/s*. There was good agreement between the Lagrangian EPIC computations (with the 2D conversion algorithm), the Eulerian CTH computations, and the test data.

A more complex and challenging example is shown in Figure 5, where a long rod (tungsten or molybdenum) impacts a confined ceramic (silicon carbide) target (Holmquist and Johnson, 2002). The upper portion shows a comparison of the test data and the computed results. For a tungsten rod at an impact velocity of 1410 *m/s* the rod experiences interface defeat; it does not penetrate the ceramic but rather moves radially outward between the top surface of the ceramic and the bottom surface of the steel plug. For a slightly higher velocity of 1645 *m/s* the rod dwells on the surface of the ceramic for about 18 μs, and then begins to penetrate. The highest velocity of 2175 *m/s* penetrates without any dwell. The understanding of these phenomena (interface defeat, dwell, penetration) are important for the design of ceramic armor. The computations show excellent agreement with the test data. Both the numerical algorithms and the material models must be robust and accurate to achieve these results.

The computed response of the interface defeat computation (1410 *m/s*) is shown in the center portion of Figure 5 and the computed response for the dwell and penetration computation (1645 *m/s*) is shown in the lower portion. The 2D axisymmetric elements are automatically converted into particles as they become highly distorted. For the lower velocity (1410 *m/s*) the localized region of damage in the target does not continue to grow and the damage is not sufficient to cause extensive failure of the ceramic. For the increased velocity (1645 *m/s*) the localized region of damage (under the surface of the ceramic) continues to grow, and at about 18 μs after impact the ceramic is damaged enough to cause the dwell to cease and the penetration to begin.

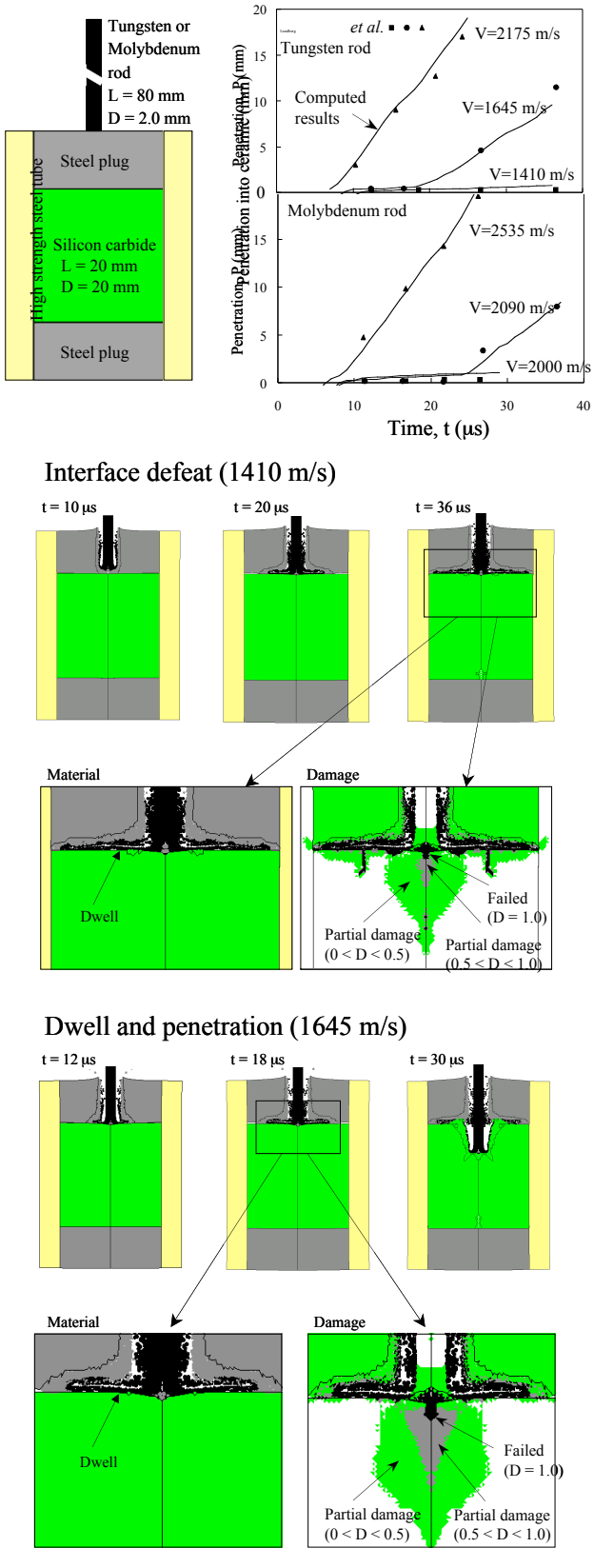


Fig. 5, Computations and test data for interface defeat, dwell and penetration into a confined ceramic target

The next example, in Figure 6, demonstrates the capability of the 3D computational approach. Here the projectile has a tungsten core with a copper sleeve, and the target is composed of three plates (aluminum, mild steel, steel) followed by a large air space and an aluminum witness plate. The length of the projectile is 127 *mm*, the impact velocity is 2000 *m/s* and the obliquity is 45 *degrees*. This example is not intended to represent a real problem, but rather to illustrate a computational capability for a complex problem. The upper portion of Figure 6 shows a view of the plane of symmetry at 100 μs and it can be seen that many of the elements have been converted into particles. There is contact and sliding between the three plates, and between the copper sleeve and the tungsten core. Some of the particles are attached to the adjacent finite elements, some are sliding on the finite elements, some are in fragments containing multiple particles, some are interacting with particles of different materials, and some are simply traveling freely through space. The center portion shows a different view at 200 μs, and it can be seen that the projectile and some of the Behind Armor Debris (BAD) has reached the aluminum witness plate. The lower portion shows damage at 200 μs, with red indicating that the material is fully damaged (it cannot develop shear or tensile stresses). This approach does not allow for the direct representation of fragment sizes smaller than the individual particle sizes, but it can be seen that some of the fragments are larger and are composed of multiple particles that are not fully damaged.

For the fully damaged (failed) particles, however, it is possible to compute the sizes of the fragments that are smaller than the particle size by use of a computational fragment algorithm (Johnson et.al., 1990). This algorithm computes the size as a function of the history of the equivalent stress, density and strain rate. The computational implementation is an extension of the original work of Grady, 1987.

The visualization of the computations in Figure 6 is performed with the Presto Visualizer. In addition to its capabilities noted previously, a unique and effective approach is used to visualize the particles. As seen in Figure 6, EPIC results may have thousands of particles that need to be visualized interactively in 3D using OpenGL. There are several options available to visualize particles, including single point sources (each sphere represented by a dot) or actual polygons used to represent the geometry of all individual spheres. A point source visualization of particles would not look like spheres, while representing each sphere individually as polygons would slow the interactivity due to the many millions of 3D polygons that may have to be drawn. In Presto, neither point sources nor polygon spheres are used. Instead a single square polygon is positioned at the location of each particle and pointed in the direction of the OpenGL camera, and the size of each square polygon corresponds to the radius of that particle.

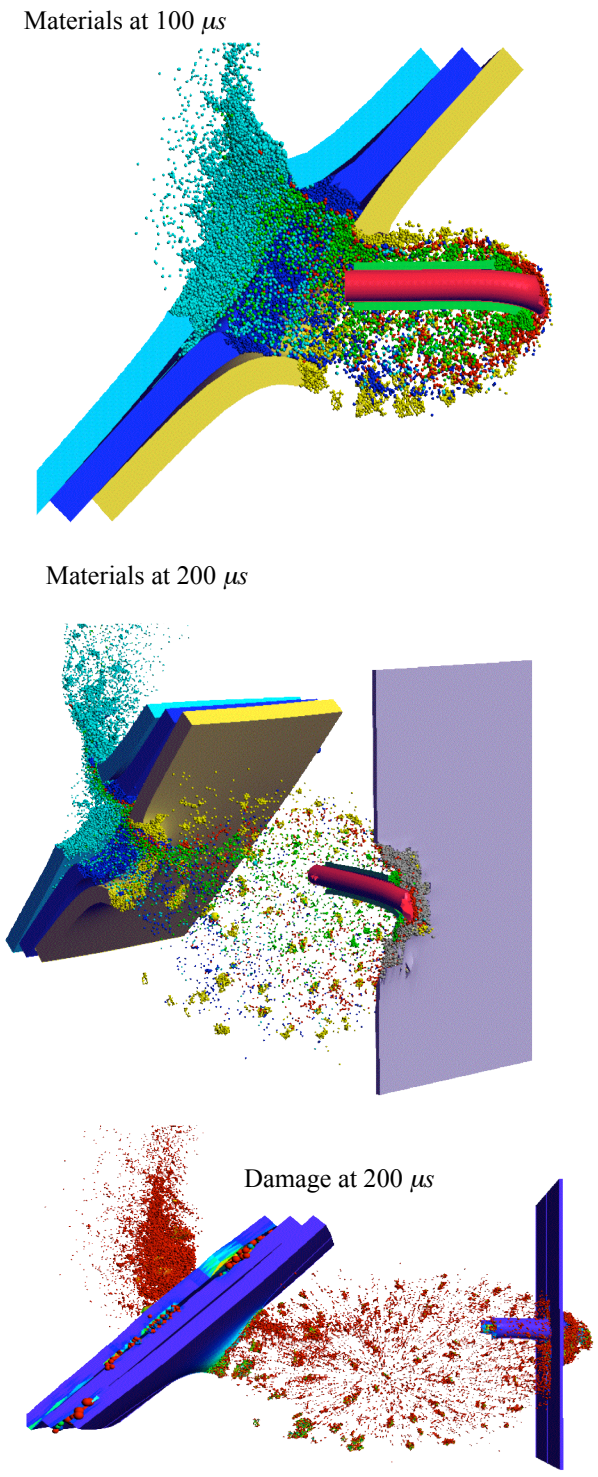


Fig. 6, Complex computation of a projectile impacting a multi-plate target

Using OpenGL's texture map features, an image of a sphere is drawn on top of each of the square polygons. Since each sphere looks the same, only a single texture map image of the sphere is required, and this image is generated on-the-fly based on the location of the light

source. The result is high interactivity (significantly lower number of polygons) and high quality (a high resolution image of the sphere texture map is used). By displaying spherical particles in this way, results with thousands or millions of particles can be interactively visualized on modest desktop systems.

The next example, in Figure 7, illustrates the effect of impact velocity for a long tungsten rod (127 *mm*) perforating a steel plate at an obliquity of 45 *degrees*. The upper three responses in Figure 7 are for impact velocities of 1000, 2000 and 3000 *m/s*. The plane of symmetry is shown such that details of the responses can be seen. The times at which the responses are shown are inversely proportional to the impact velocities (300, 200 and 100 μs, respectively). The velocities of the fragments and projectiles are approximately proportional to the impact velocities. Although the fragment velocities for the three cases may appear to be similar to one another because of the similar spacial distributions in Figure 7, they are not similar because the difference in times requires a difference in velocities. The lower response shows the entire problem (without a plane of symmetry) and the distribution of damage for the 3000 *m/s* velocity.

Some general comments can be made about the results in Figure 7. First, the residual velocity, for the lowest impact velocity (1000 *m/s*), is reduced more than the other two cases as determined by the positions (note that the initial impact velocities multiplied by the corresponding times shown in Figure 7 give identical free flight travel distances). Also, all three cases provide a clockwise (nose down) rotation of the projectile as it perforates the target, and this is due to the projectile sensing the rear free surface as the path of least resistance. A related observation is that more of the eroded tungsten material appears on the upper side of the target hole when compared to the lower side. This is probably due to the increased target resistance on the upper side, when compared to the lesser resistance on the lower side that is nearer the free surface.

Perhaps the greatest difference between the three cases is that the hole diameter in the target increases significantly as the impact velocity increases. For the 1000 *m/s* impact the hole diameter is slightly larger than the rod diameter. For the 3000 *m/s* impact, however, the hole diameter is several times greater than the rod diameter. A closely related result is that there is much more target mass distributed behind the target for the higher impact velocities. Although it is difficult to quantify, it appears that the fragments for the lowest velocity are fewer in number and larger in size. The higher velocity impacts have a range of smaller fragments (single particles) to larger fragments (groups of particles). A common characteristic for all three cases is the formation of lower-velocity fragments that are formed from the rear surface of the target around the outer portion of the hole. These same general trends have also been observed experimentally.

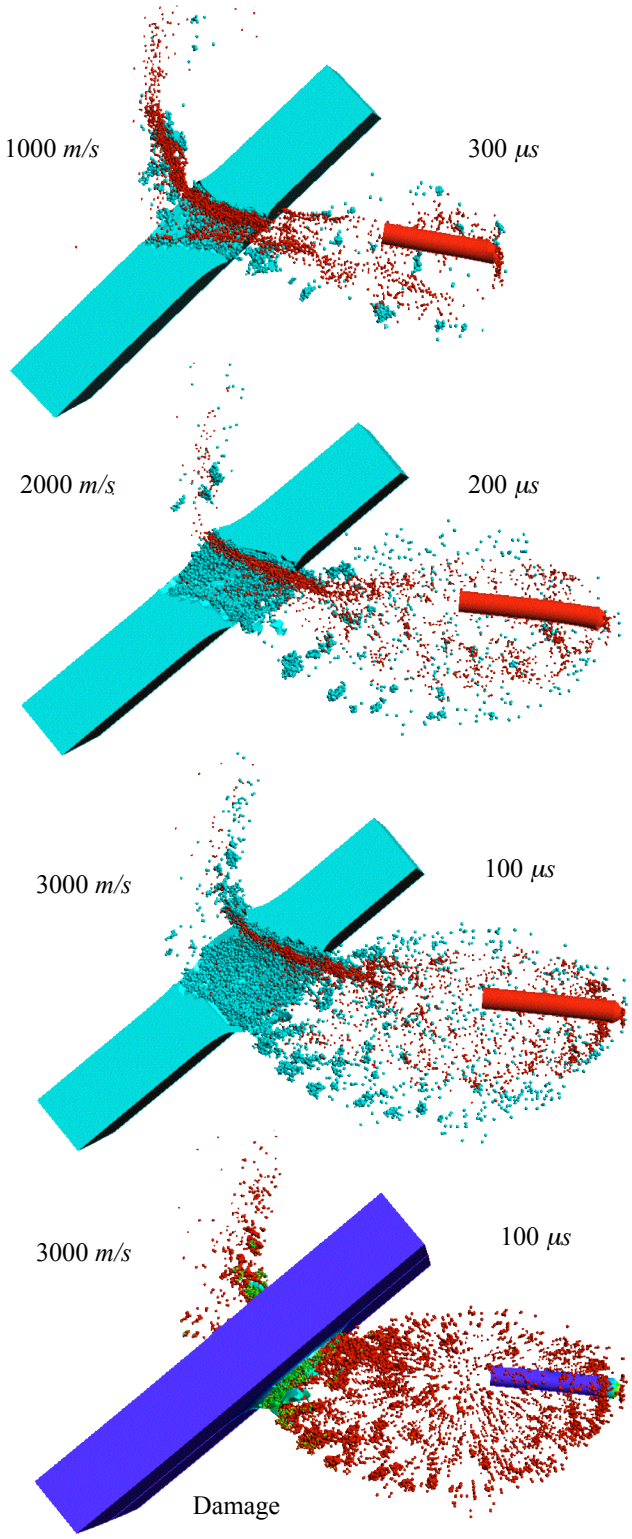


Fig. 7, Computations of a tungsten rod perforating a steel plate at three impact velocities

The example in Figure 8 consists of a long tungsten rod (127 *mm*) impacting a target composed of a ceramic (silicon carbide) plate over an aluminum plate, followed by a large space and a thin aluminum witness plate. The impact velocity of 1500 *m/s* again provides a large overmatch condition such that the BAD can be generated. All three views are shown at the same time (400 μs) after impact. In this example cracks form in the brittle ceramic plate, unlike the ductile response of the metallic target plates shown previously. Also, it can be seen that the BAD forms a large hole in the witness plate.

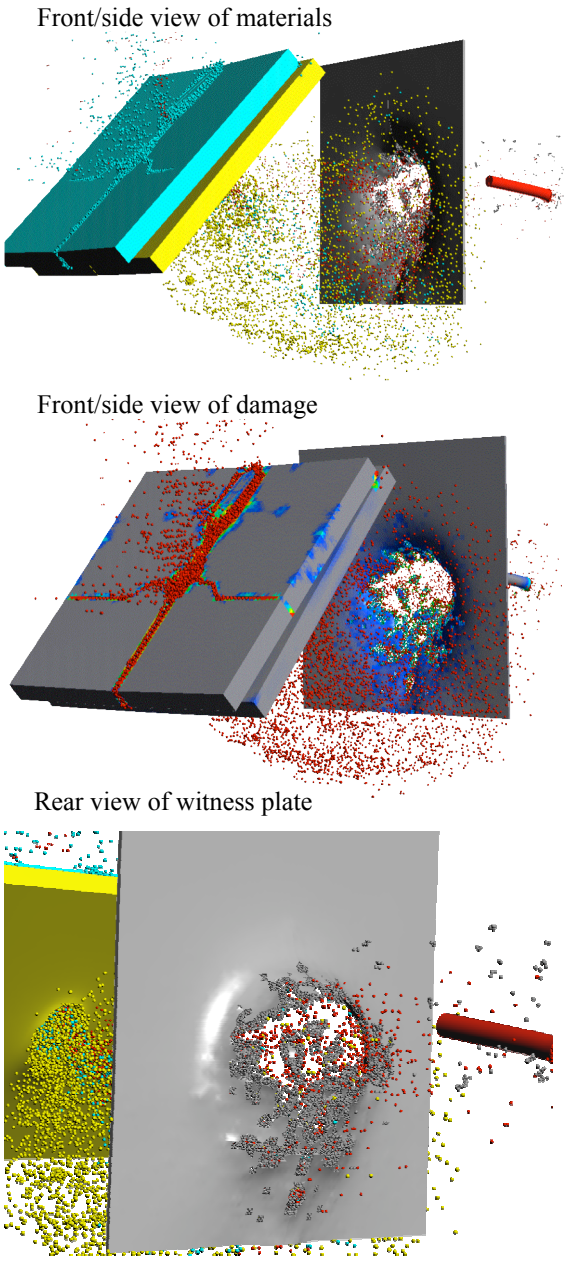


Fig. 8, Computation of a projectile impacting a target with a ceramic component

The final example is shown in Figure 9 and it is representative of a small caliber (8.62 *mm* diameter) projectile impacting a layered ceramic/metallic component (silicon carbide over aluminum) similar to that used in some body armors. Here the normal impact velocity is 800 *m/s* and it impacts 5.1 *mm* below the center of the square tile (51 *mm* x 51 *mm*).

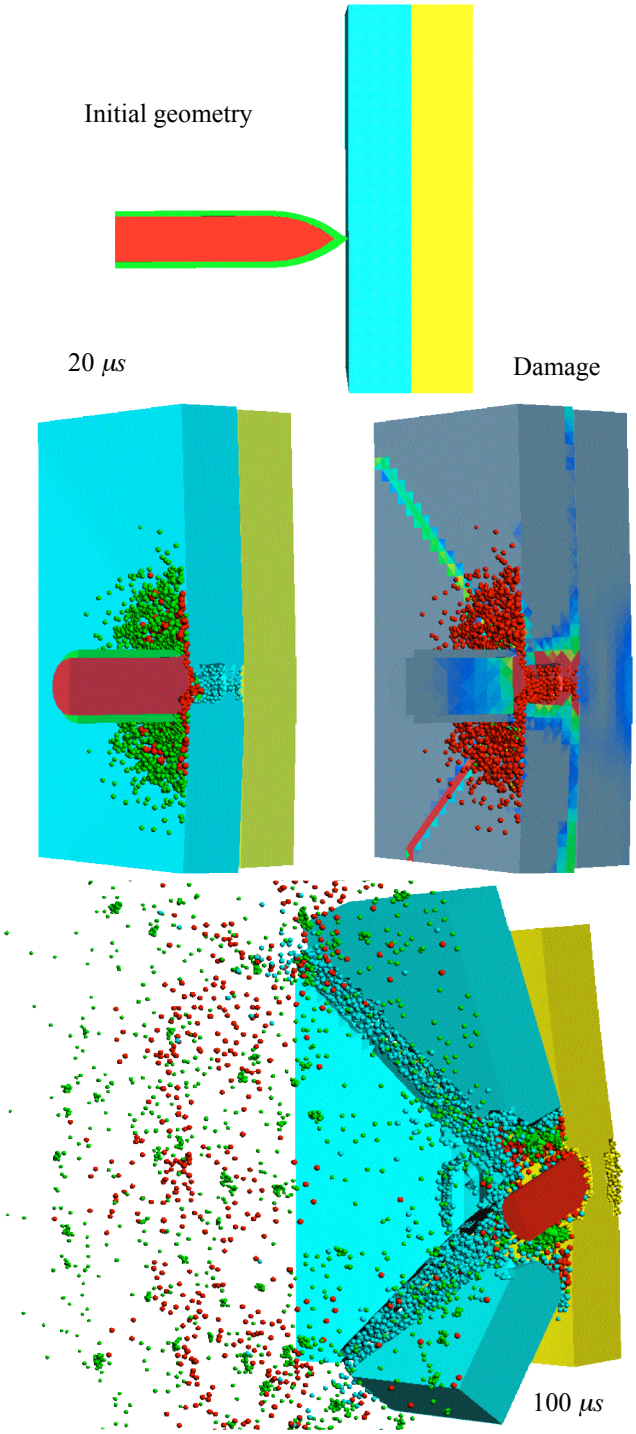


Fig. 9, Computation of a small caliber projectile impacting a body armor component

The upper portion of Figure 9 shows the copper jacket around the hard steel core in the initial geometry. At 20 μs after impact the ceramic plate has been significantly damaged (under the projectile and along two radial cracks) even though there is no significant penetration into the ceramic. This dwell is similar to that shown previously in Figure 5, with the eroded tip of the projectile (copper jacket and steel core) moving radially outward along the top surface of the ceramic plate. For this case, however, the finite thickness of the ceramic plate does not provide enough resistance to defeat the projectile. Later, at 100 μs after impact, the steel core has penetrated the ceramic and some of the aluminum, the ceramic has broken into three distinct pieces (for the half of the problem shown), and the copper jacket has moved forward along the core. The copper has much less strength than the steel core and is therefore not able to decelerate itself as much as the stronger steel core. At this point the steel core and the aluminum are moving at essentially the same low velocity and the core does not perforate the aluminum.

4. SUMMARY AND CONCLUSIONS

This article has provided an overview of the issues associated with computations for ballistic impact problems. It has also presented some recent developments that have significantly improved the capabilities in this technology, including an algorithm to automatically convert distorted elements into particles. Examples have been provided to show agreement with test data and to demonstrate capabilities for complex 3D computations.

ACKNOWLEDGEMENTS

The research reported in the article was performed in connection with contract DAAD19-03-D-0001 with the U. S. Army Research Laboratory. The views and conclusions contained in this article are those of the authors and should not be interpreted as presenting the official policies or positions, either expressed or implied, of the U. S. Army Research Laboratory or the U. S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U. S. Government is authorized to reproduce and distribute preprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

Grady, D., 1987: Fragmentation of rapidly expanding jets and sheets, *Int. J. Impact Eng.,* **5,** 285-292.

Holmquist, T. and Johnson, G., 2001: Response of silicon carbide to high velocity impact, *J. Appl. Phys.,* **91,** 5858-5866.

Johnson, A., 2004: Presto Visualizer 2.0, Parallel scientific visualization of remote datasets, User Guide.

Johnson, A. and Quammen, C., 2003: Large scale scientific visualization on Cray MPP architectures, Proceedings of the 45th Cray User Group (CUG) Conference, Columbus, Ohio.

Johnson, G., Beissel, S. and Stryk, R., 2002: An improved Generalized Particle Algorithm that includes boundaries and interfaces, *Int. J. Numer. Meth. Eng.,* **53,** 875-904.

Johnson, G. and Stryk, R., 2001: Symmetric contact and sliding interface algorithms for intense impulsive loading computations, *Comput. Meth. Appl. Mech.Eng.,* **190,** 4531-4549.

Johnson, G. and Stryk, R., 2003: Conversion of 3D distorted elements into meshless particles during dynamic formation, *Int. J. Impact Eng.,* **28,** 947-966.

Johnson, G., Stryk, R., Beissel, S. and Holmquist, T., 2002: An algorithm to automatically convert distorted finite elements into meshless particles during dynamic deformation, *Int. J. Impact Eng.,* **27,** 997-1013.

Johnson, G., Stryk., R., Holmquist, T. and Souka, O., 1990: Recent EPIC code developments for high velocity impact: 3D element arrangements and 2D fragment distributions, *Int. J. Impact Eng.,* **10,** 281-284.

Templeton, D., Holmquist, T., Meyer, H., Grove, D. and Leavy, B., 2001: A comparison of ceramic material models, Proceedings of Ceramic Armor Materials by Design Symposium, *Ceramic Transactions,* **134,** 299-308.

Advanced Computations for Ballistic Impact Problems

G. R. Johnson, S. R. Beissel, C. A. Gerlach,
R. A. Stryk, A. A. Johnson and T. J. Holmquist

Network Computing Services, Inc.

24th Army Science Conference
Nov 29 – Dec 2, 2004
Orlando, Florida

Acknowledgement

- This presentation was performed in connection with contract DAAD19-03-D-0001 with the U. S. Army Research Laboratory.

  The views and conclusions contained in this presentation are those of the author and should not be interpreted as presenting the official policies or positions, either expressed or implied, of the U. S. Army Research Laboratory or the U. S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Outline

- Motivation for ballistic impact computations
- Historical background
- Some desirable characteristics for a code
- Finite elements and meshless particles
- Examples (EPIC code)
- Summary and conclusions

Motivation for ballistic computations

- **Testing**
  - Fabrication of test hardware can be expensive, time-consuming and limited to existing materials
  - Testing can be expensive, time consuming, provide only limited data, and not be available for all parameters

- **Computations**
  - Provide detailed information
  - Can be performed quickly and inexpensively
  - Can examine conditions not attainable with testing
  - Allow for evaluation of numerous design parameters
  - Are becoming faster and more accurate

Historical background

- First ballistic computations performed at National Laboratories in 1960's

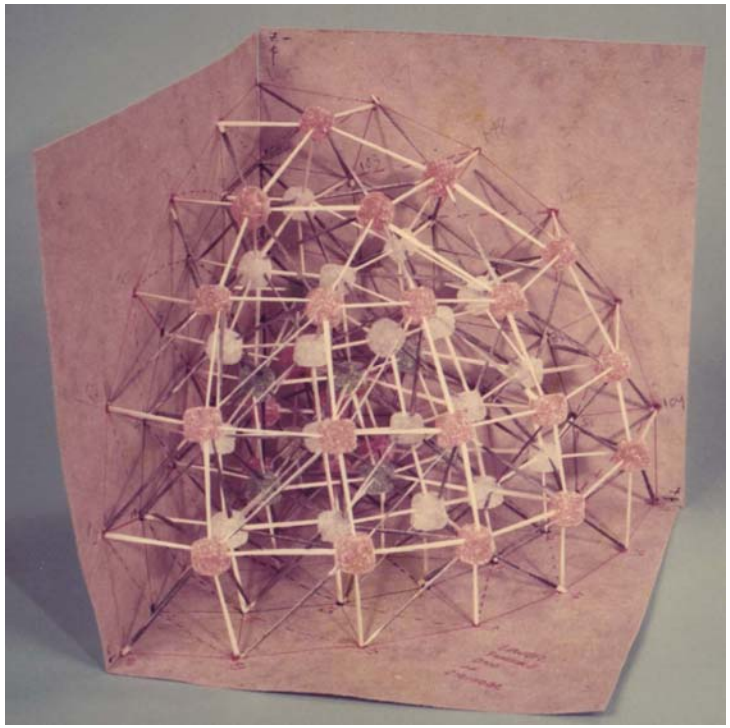
- Many challenging problems to overcome
  - High pressures, temperatures, strains and strain rates
  - Wide variety of materials
  - Failure and fragmentation of materials
  - Complex contact and sliding interactions
  - Requires significant computer capability
  - Requires sophisticated grid generation and visualization

Some desirable characteristics for a code

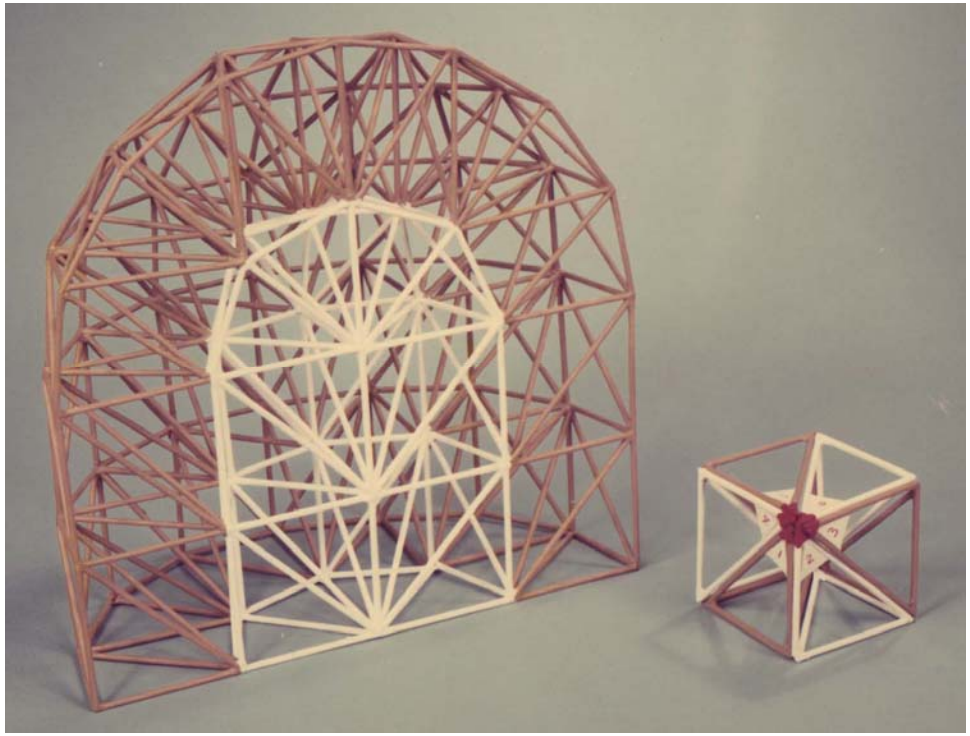
- **Robust**
  - Always runs for wide range of problems
  - Handles high pressures, temperatures, strains, strain rates
- **Accurate**
  - Proper finite element and meshless particle algorithms
  - Boundaries, interfaces and contact
  - Material models and constants for wide range of materials
- **Efficient**
  - Preprocessors, grid generators and GUIs
  - Postprocessing and visualization
  - Runs in serial and/or parallel mode

Early grid generators for the EPIC code

Original



Improved

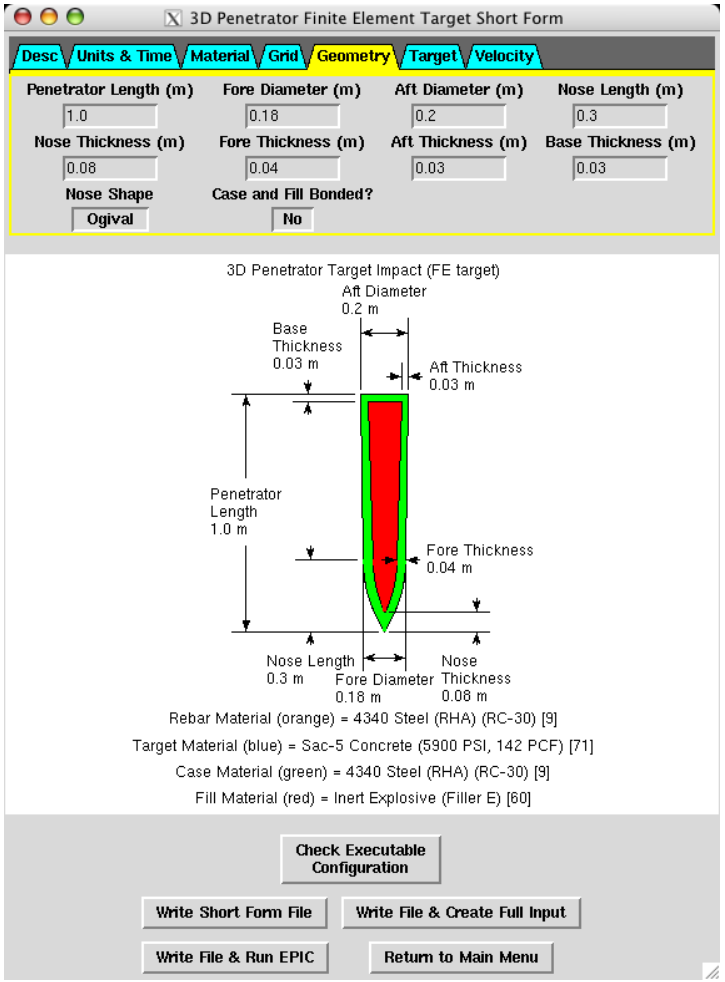


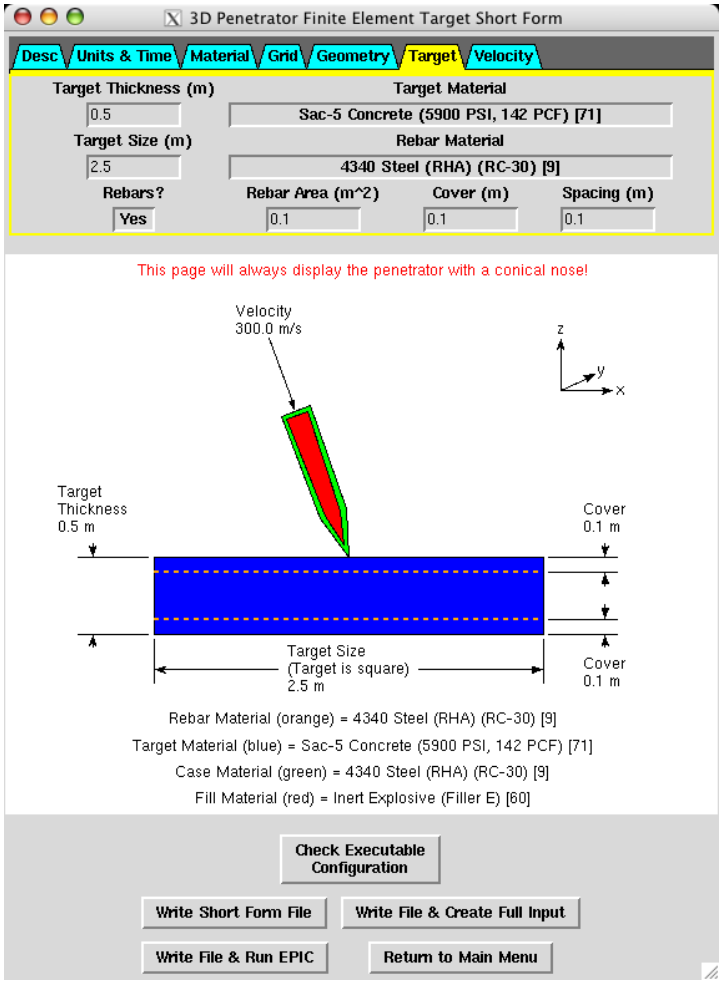
Toothpicks and gumdrops

AHPCRC

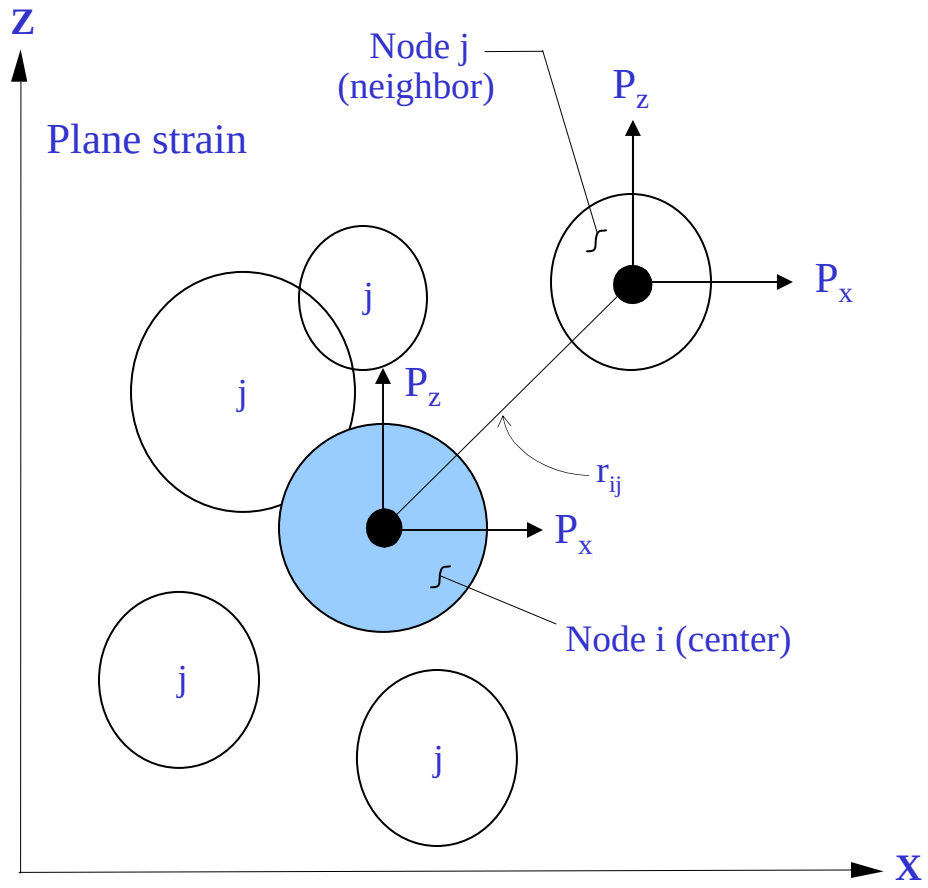
Network Computing Services, Inc.

Current Short Form GUI for EPIC code

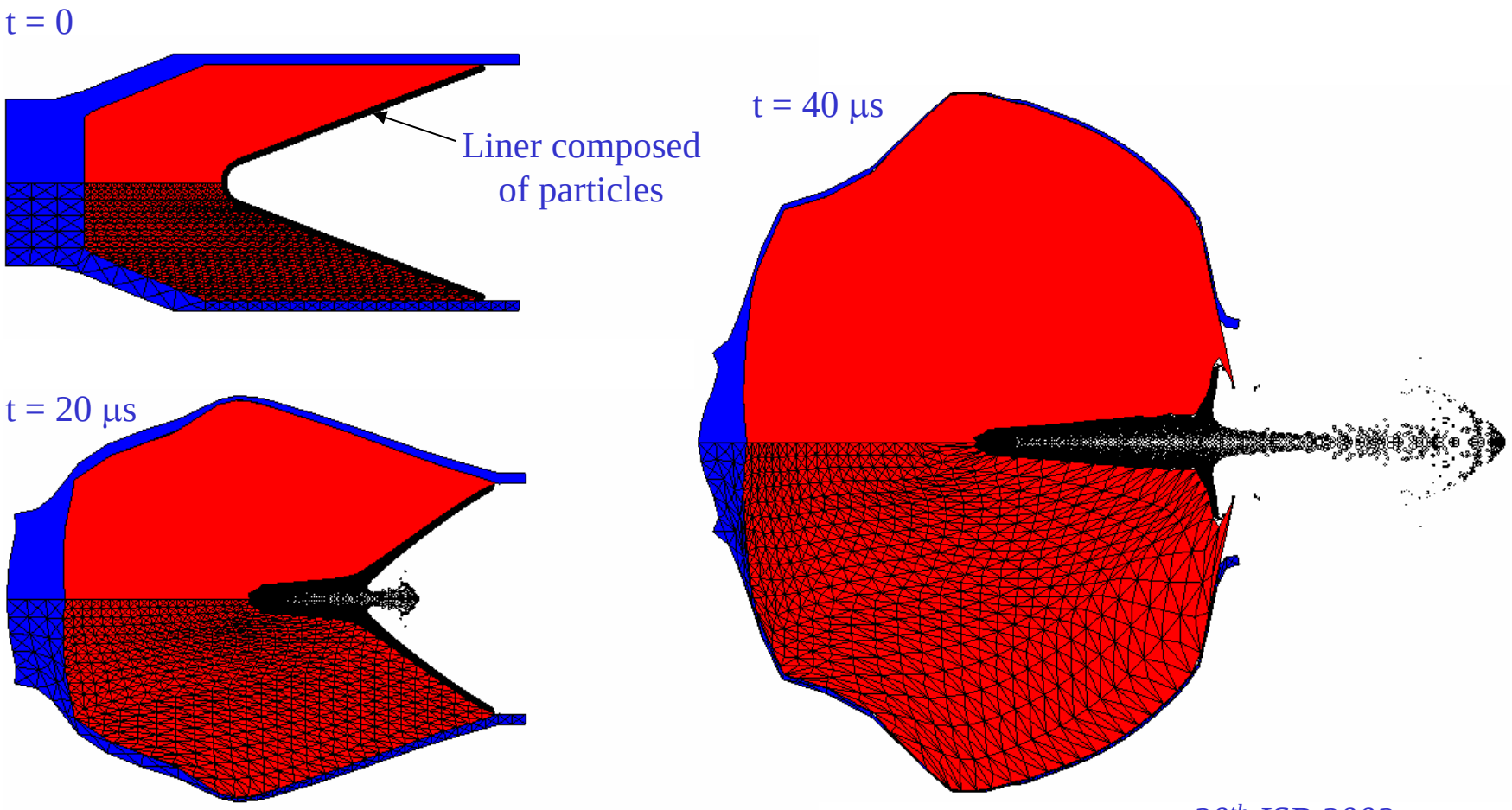




Generalized Particle Algorithm



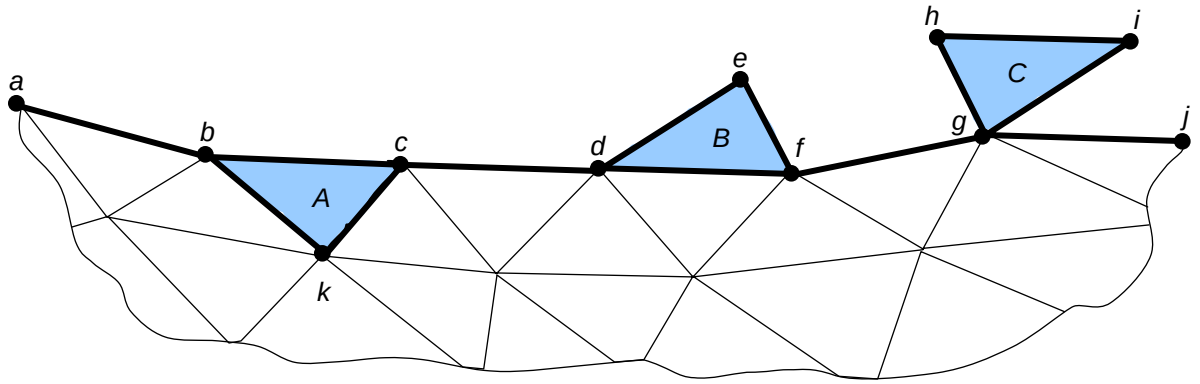
Jet formation with a shaped charge



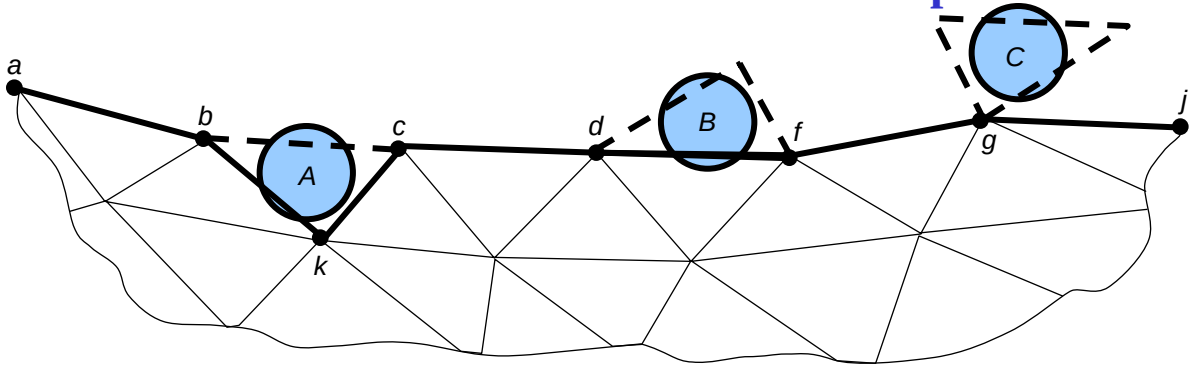
20th ISB 2002

Conversion of finite elements into particles

Interface (nodes *a* … *j*) before conversion



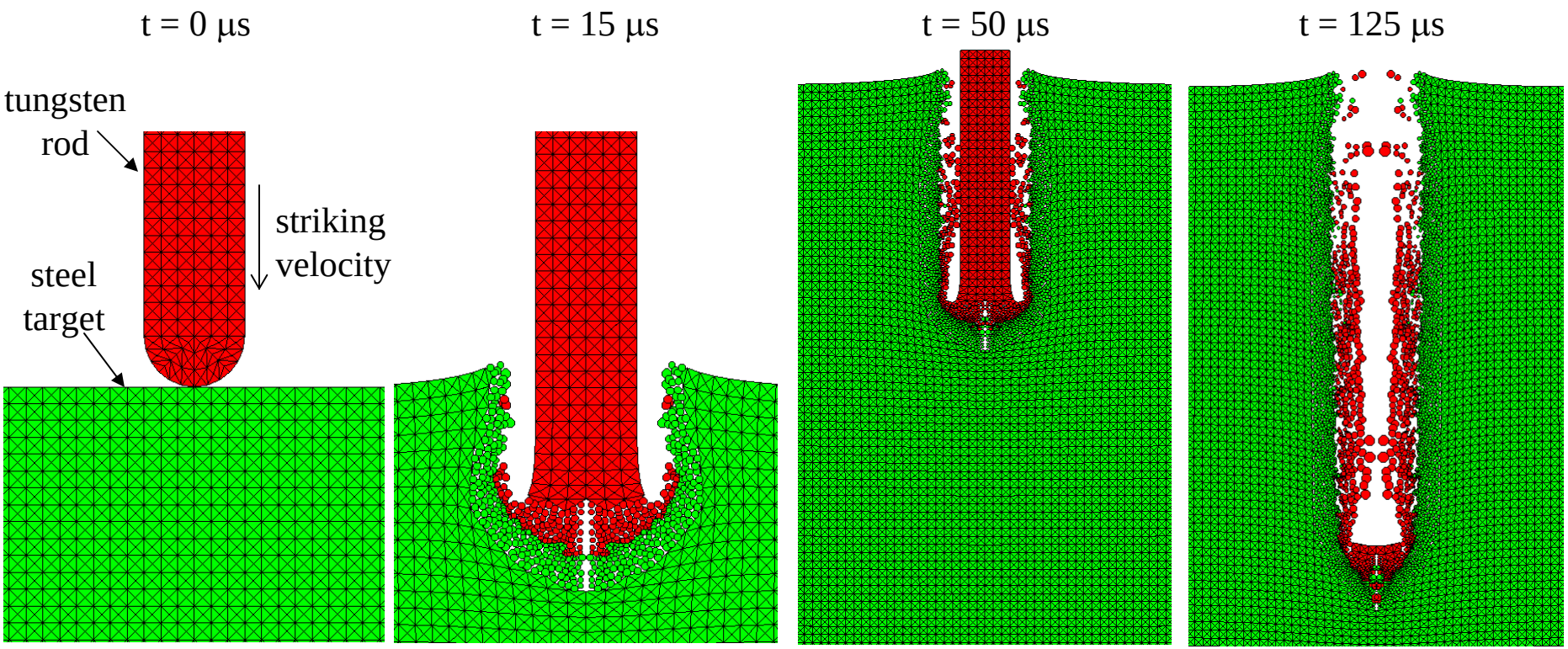
Interface after conversion of elements to particles



APS 2001

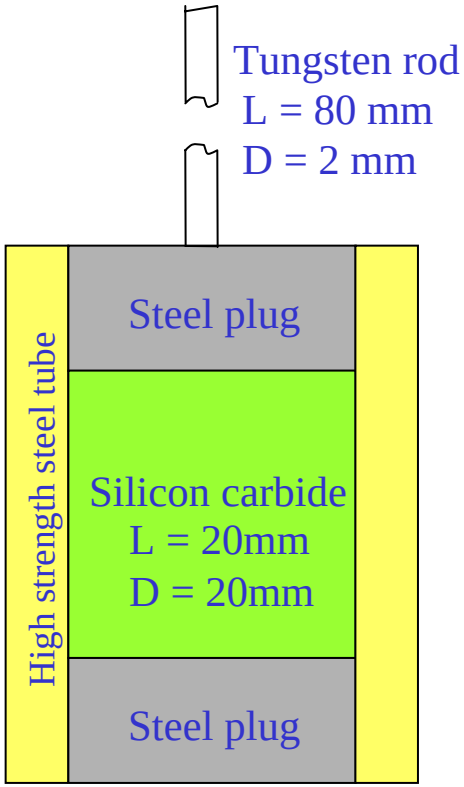
Deep penetration by long-rod penetrator

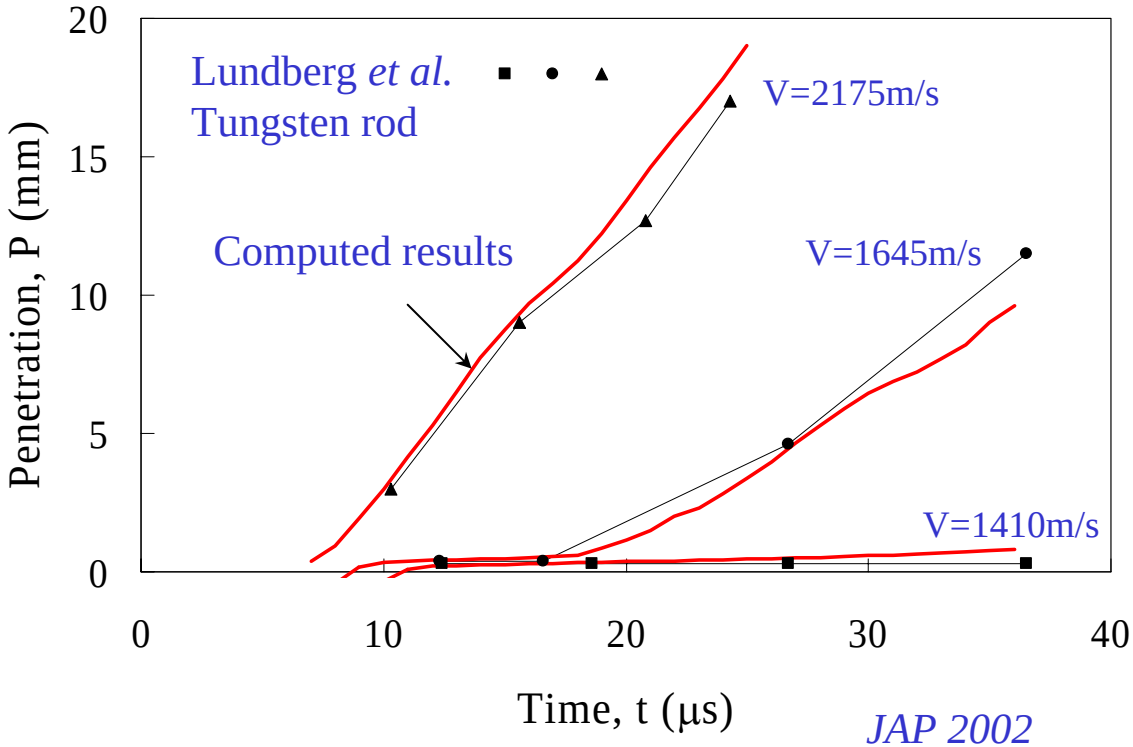
Tungsten rod penetrating steel target



APS 2001

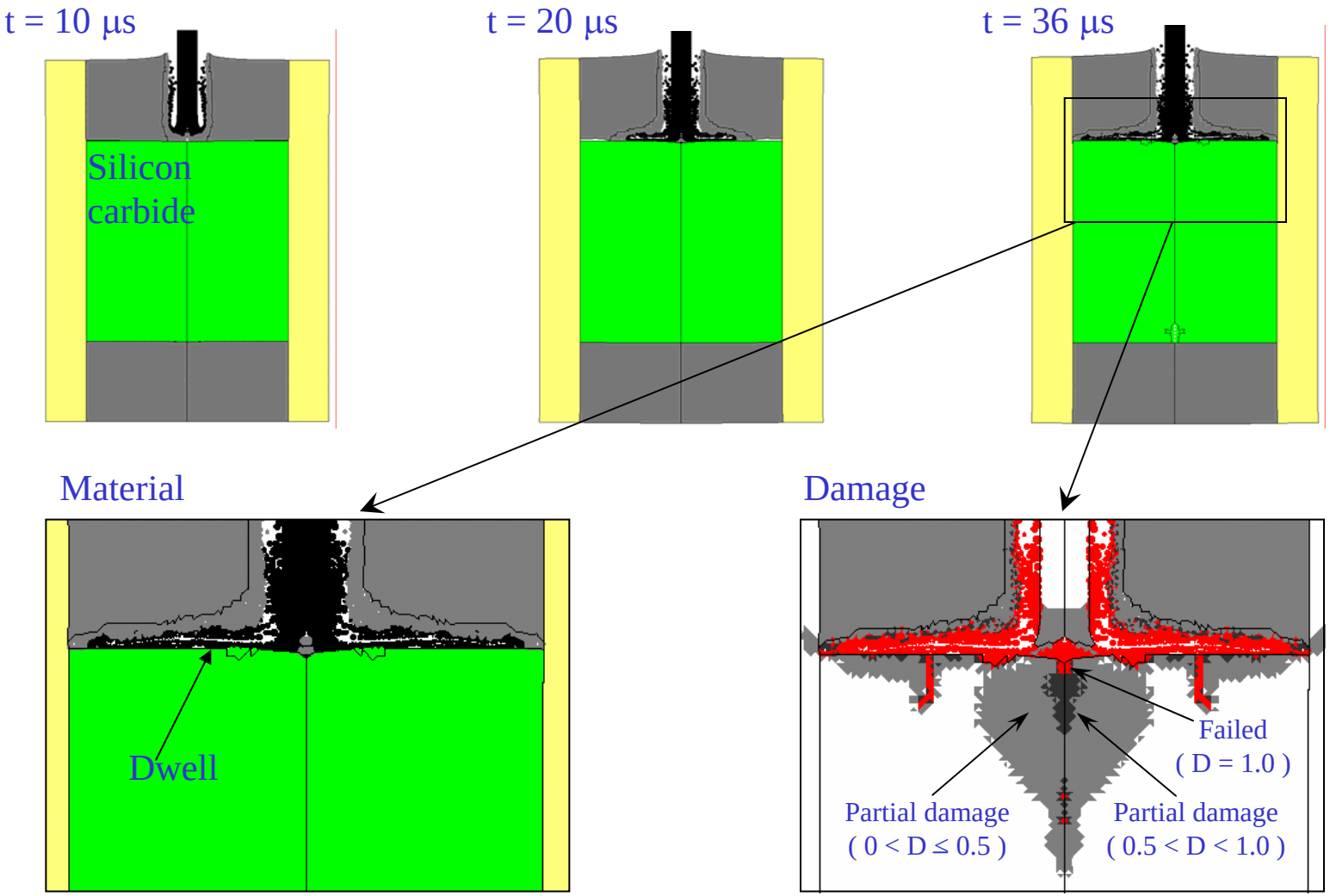
Ceramic computations and test results





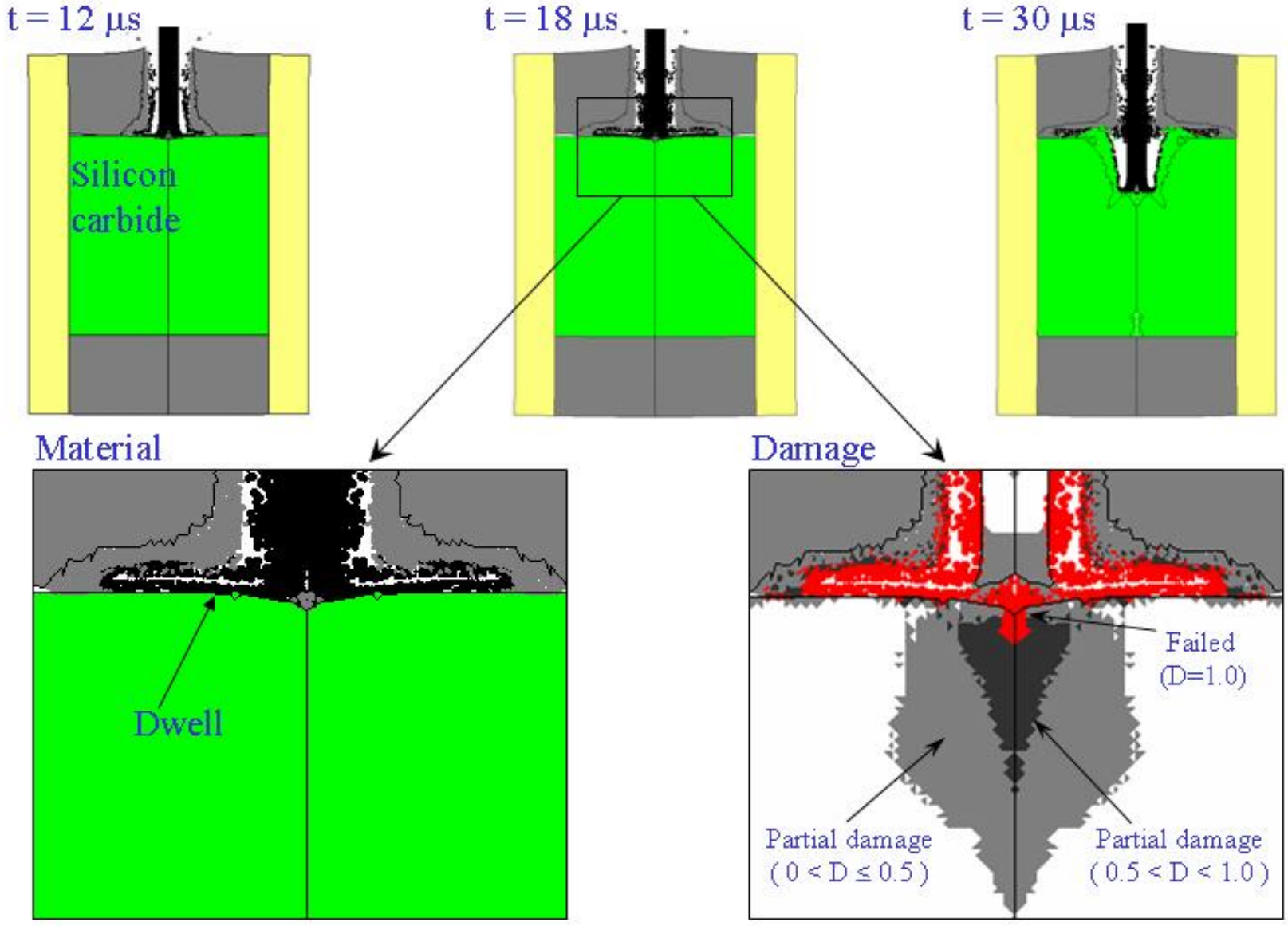
JAP 2002

Interface defeat (V_0=1410 m/s)



JAP 2002

Dwell and penetration (V_0 = 1645 m/s)



JAP 2002

AHPCRC

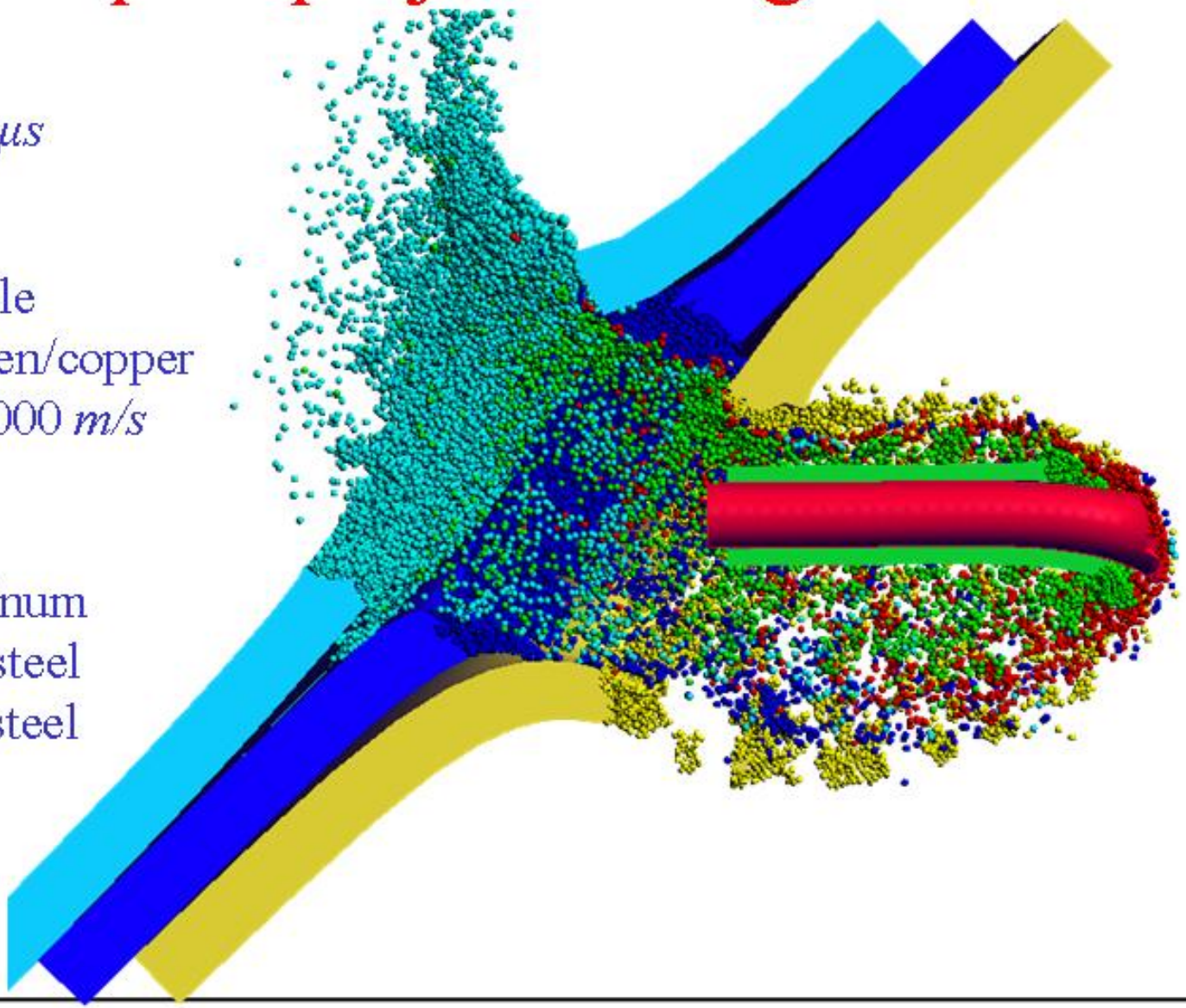
Network Computing Services, Inc.

Complex projectile-target interaction

t = 100 μs

Projectile
- tungsten/copper
- V = 2000 *m/s*

Target
- aluminum
- mild steel
- hard steel



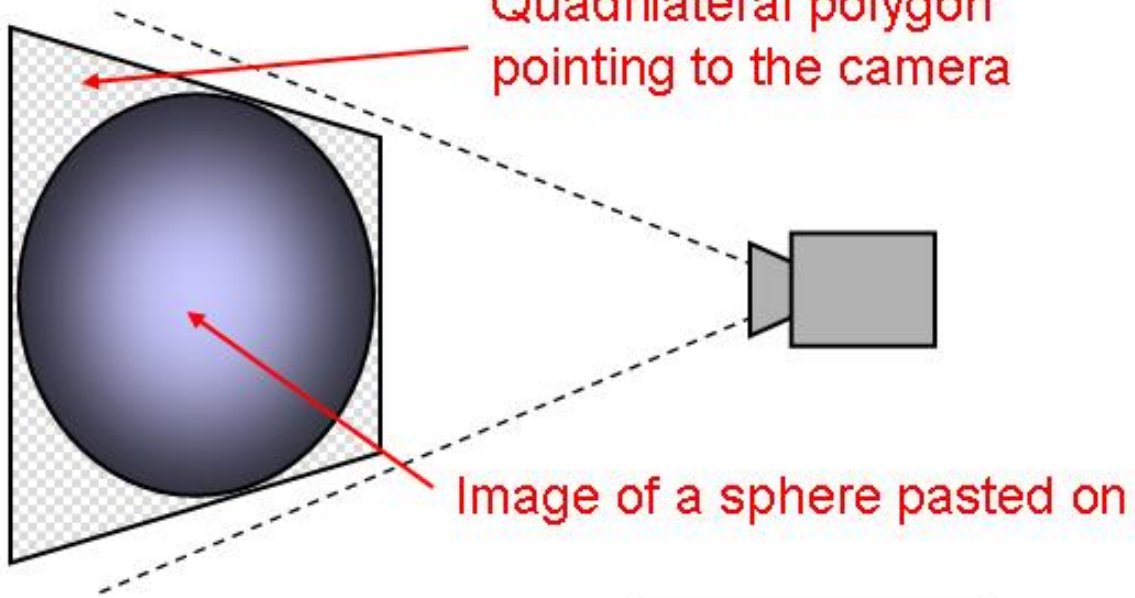
AHPCRC

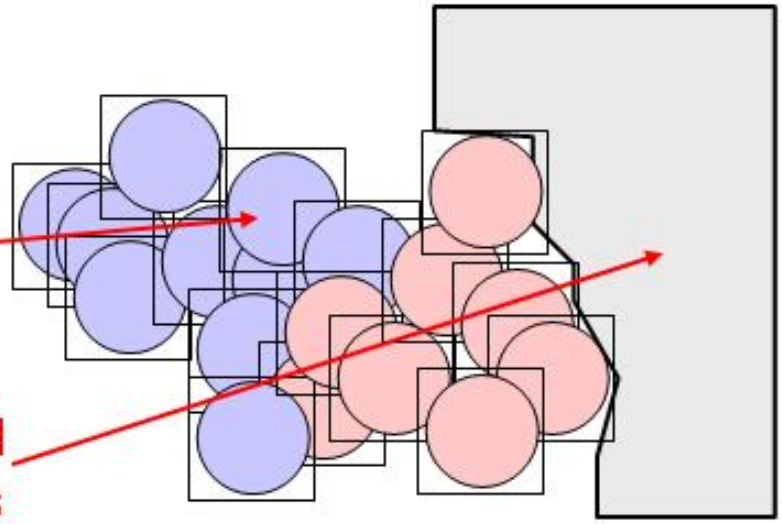
NETWORK COMPUTING SERVICES, INC.

Scientific visualization

Presto Visualizer

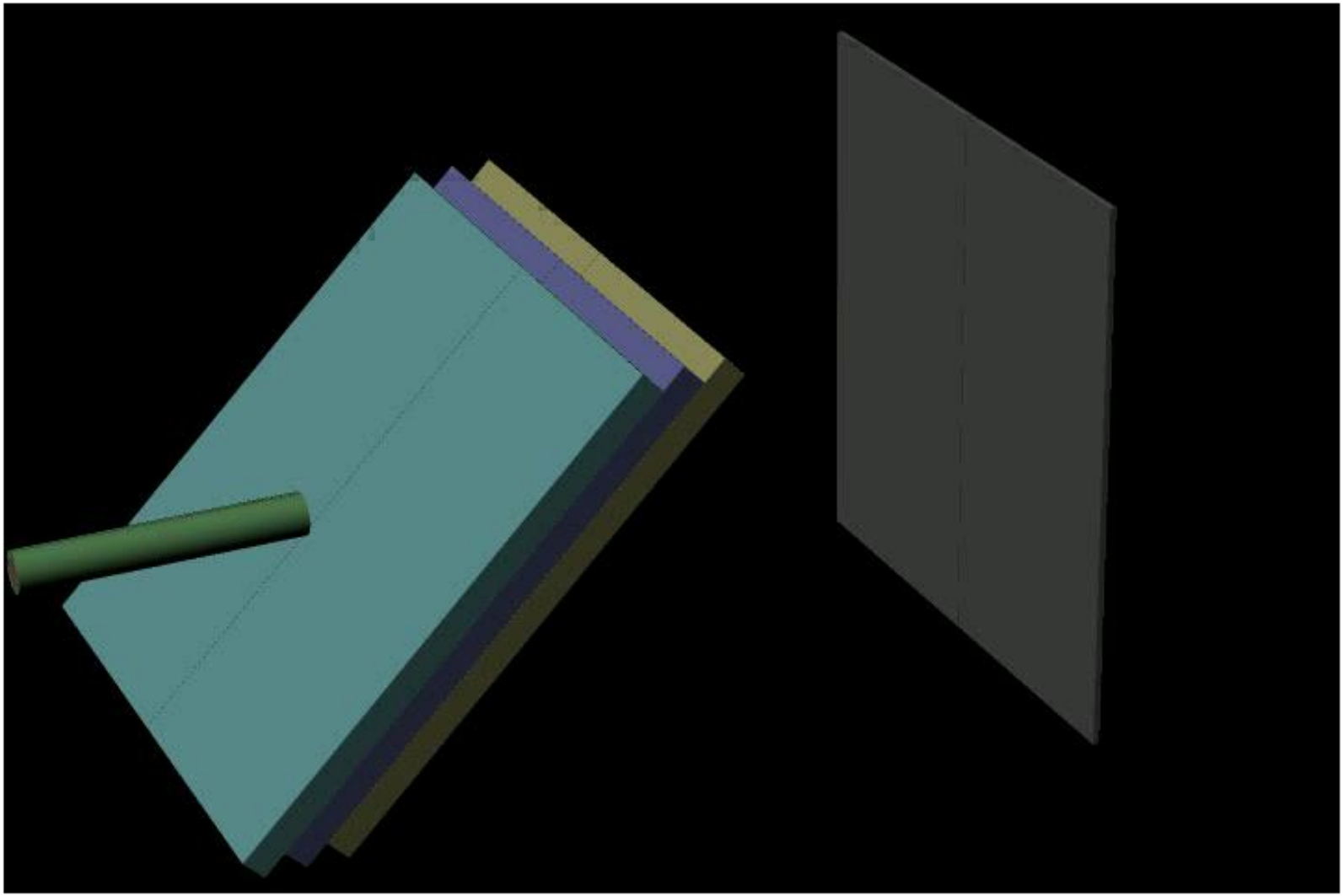
- Developed at the AHPCRC
- Client-server approach
  - Remote visualization
- Fully parallel server
  - Large data sets
- Many advanced visualization features
  - Easy-to-use
- Visualizes EPIC data sets including particles

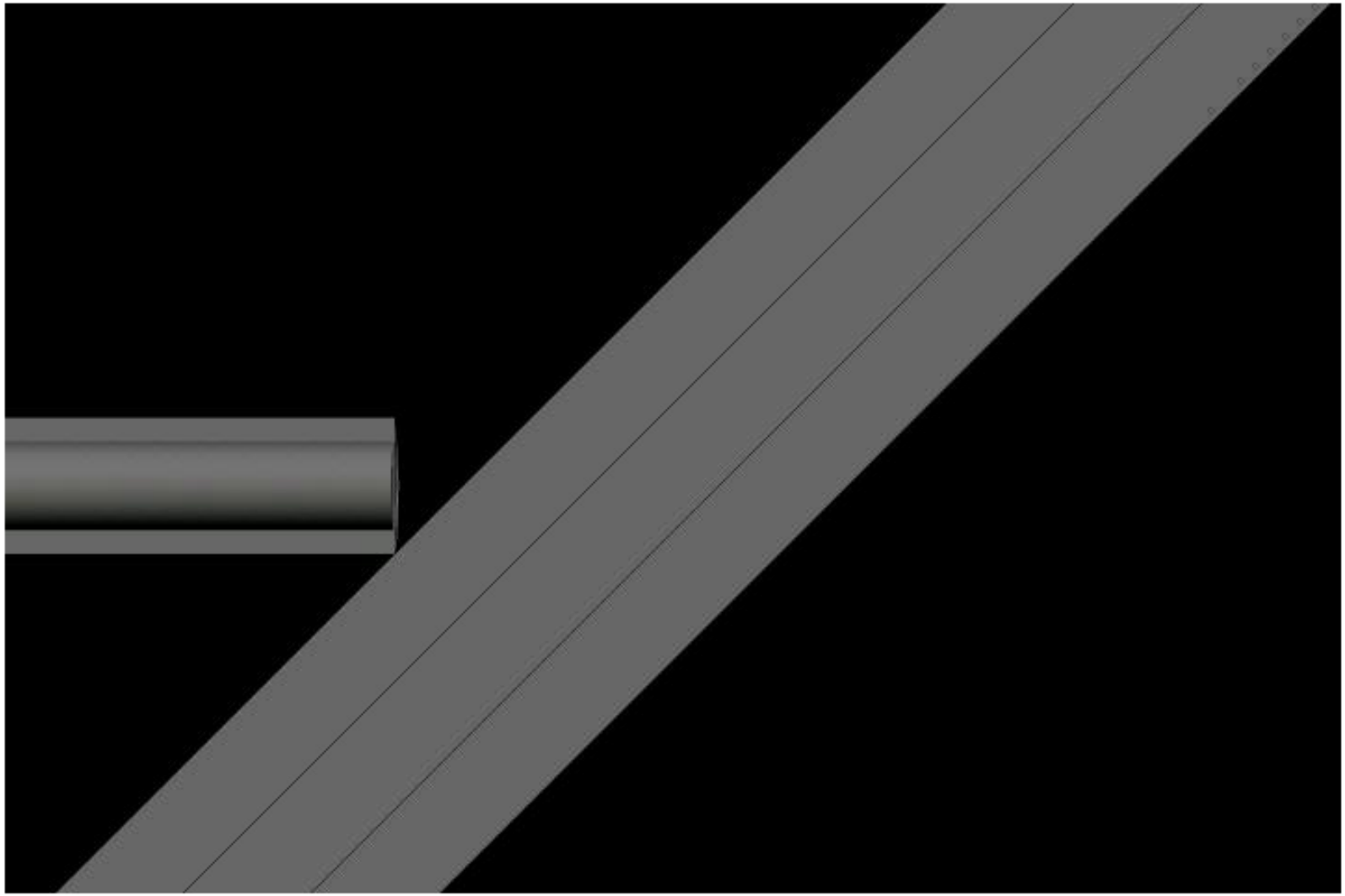




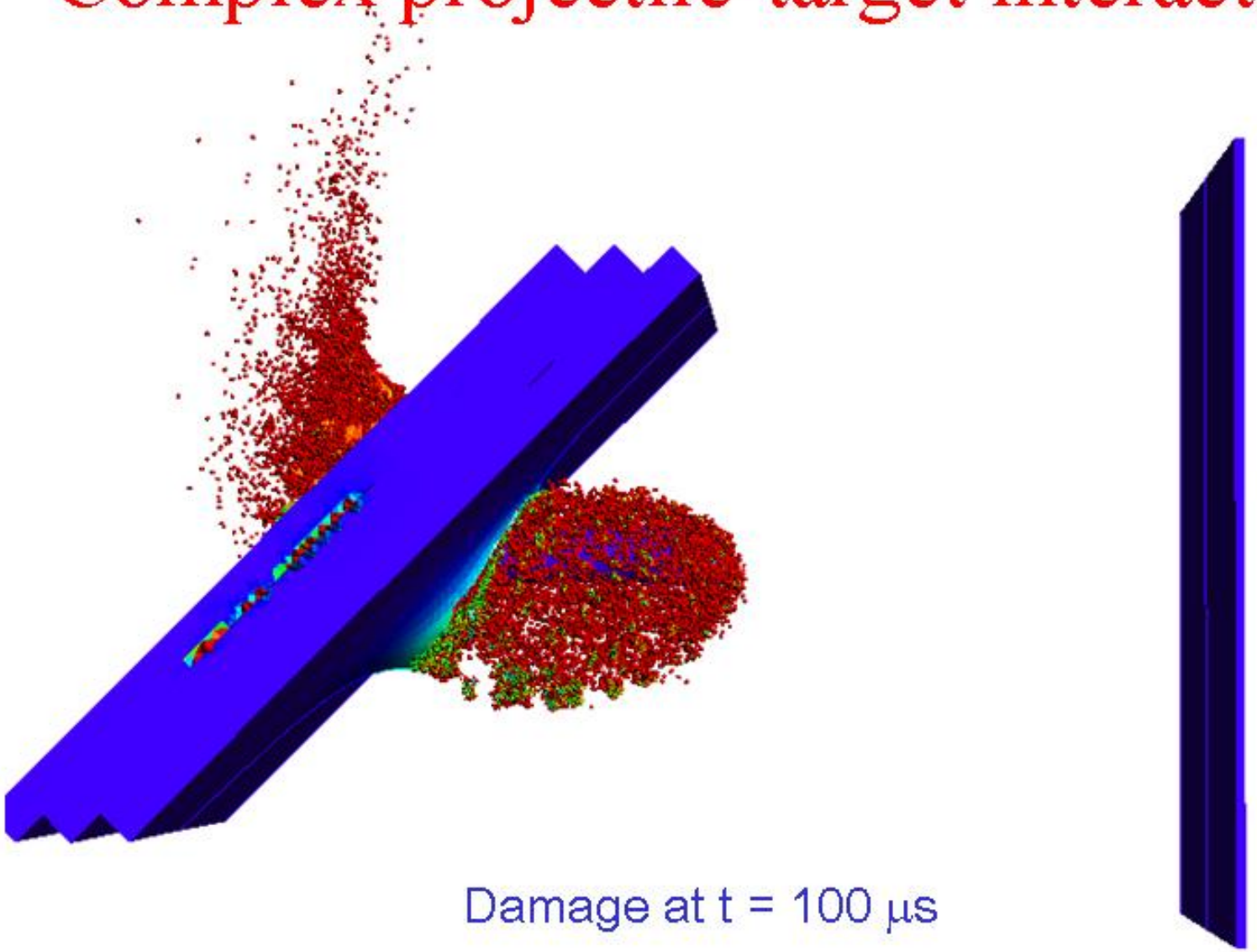
AHPCRC

NETWORK COMPUTING SERVICES, INC.



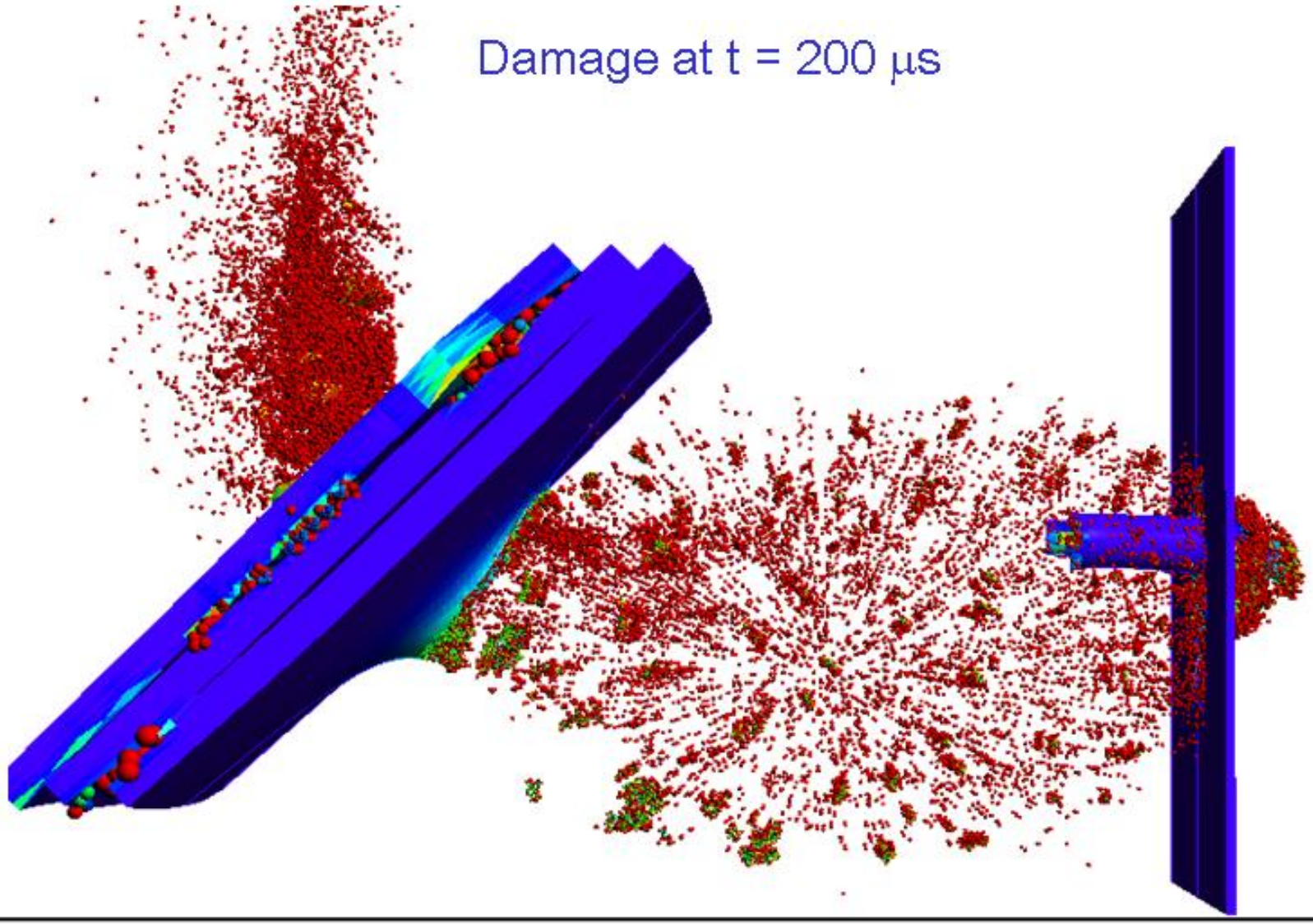


Complex projectile-target interaction



Damage at t = 100 μs

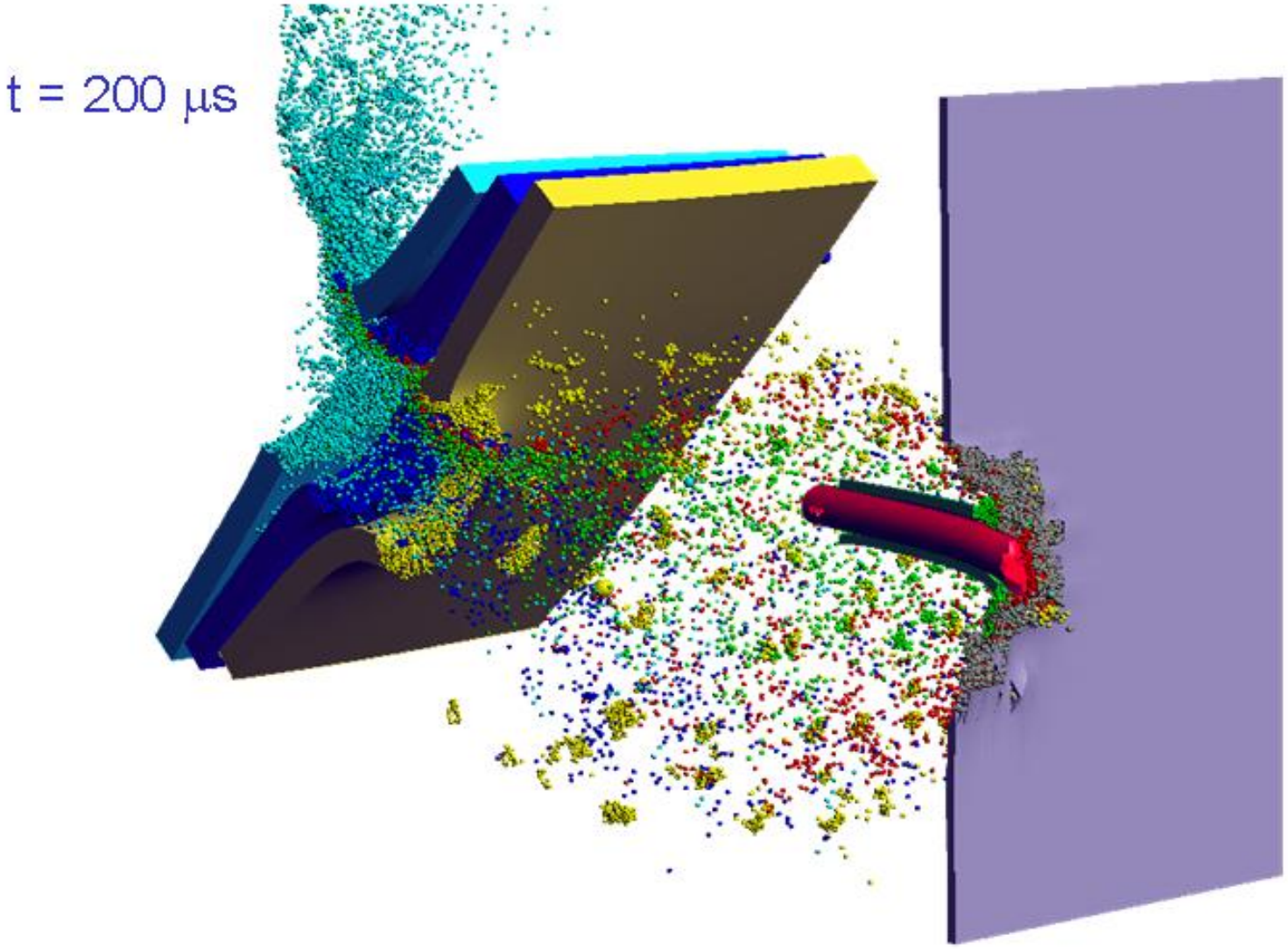
Complex projectile-target interaction



AHPCRC

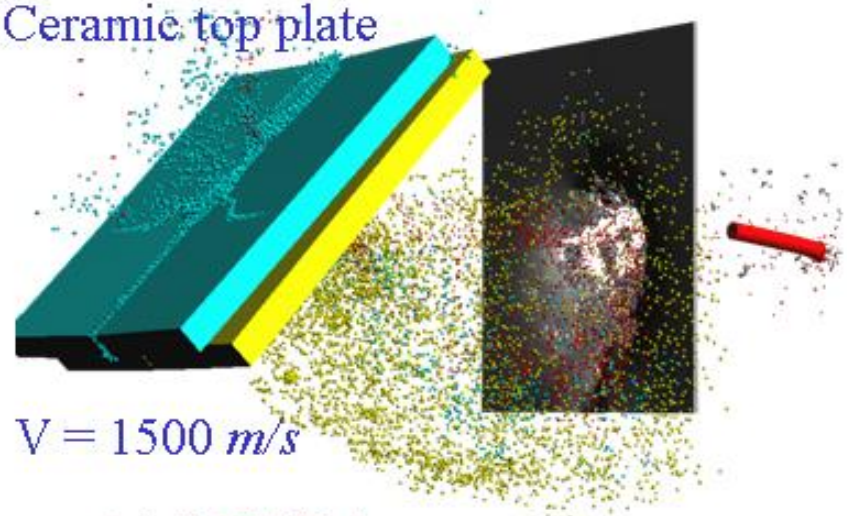
Network Computing Services, Inc.

Complex projectile-target interaction

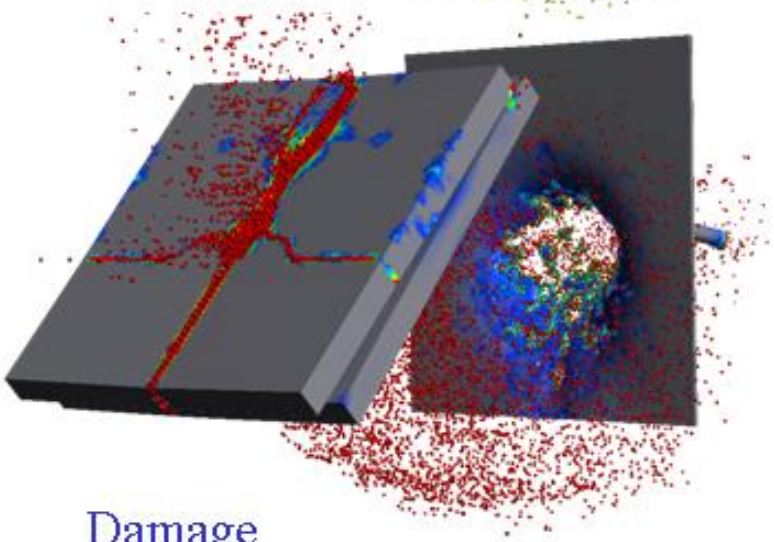


Target with ceramic component

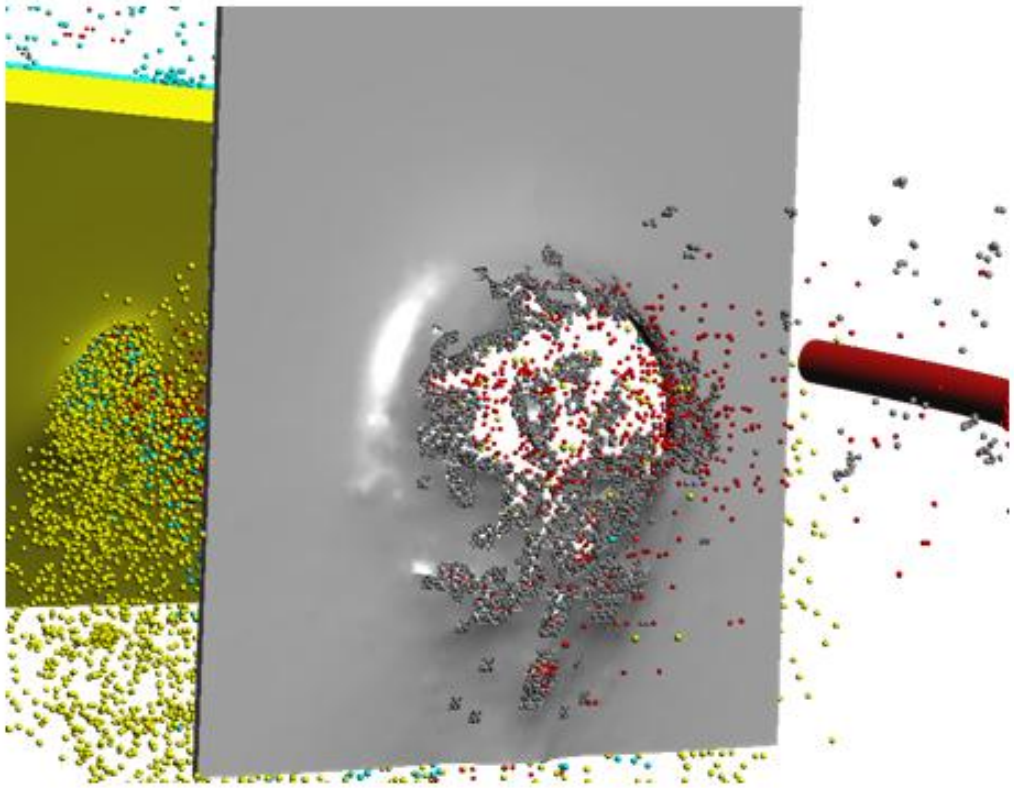
Ceramic top plate



V = 1500 *m/s*

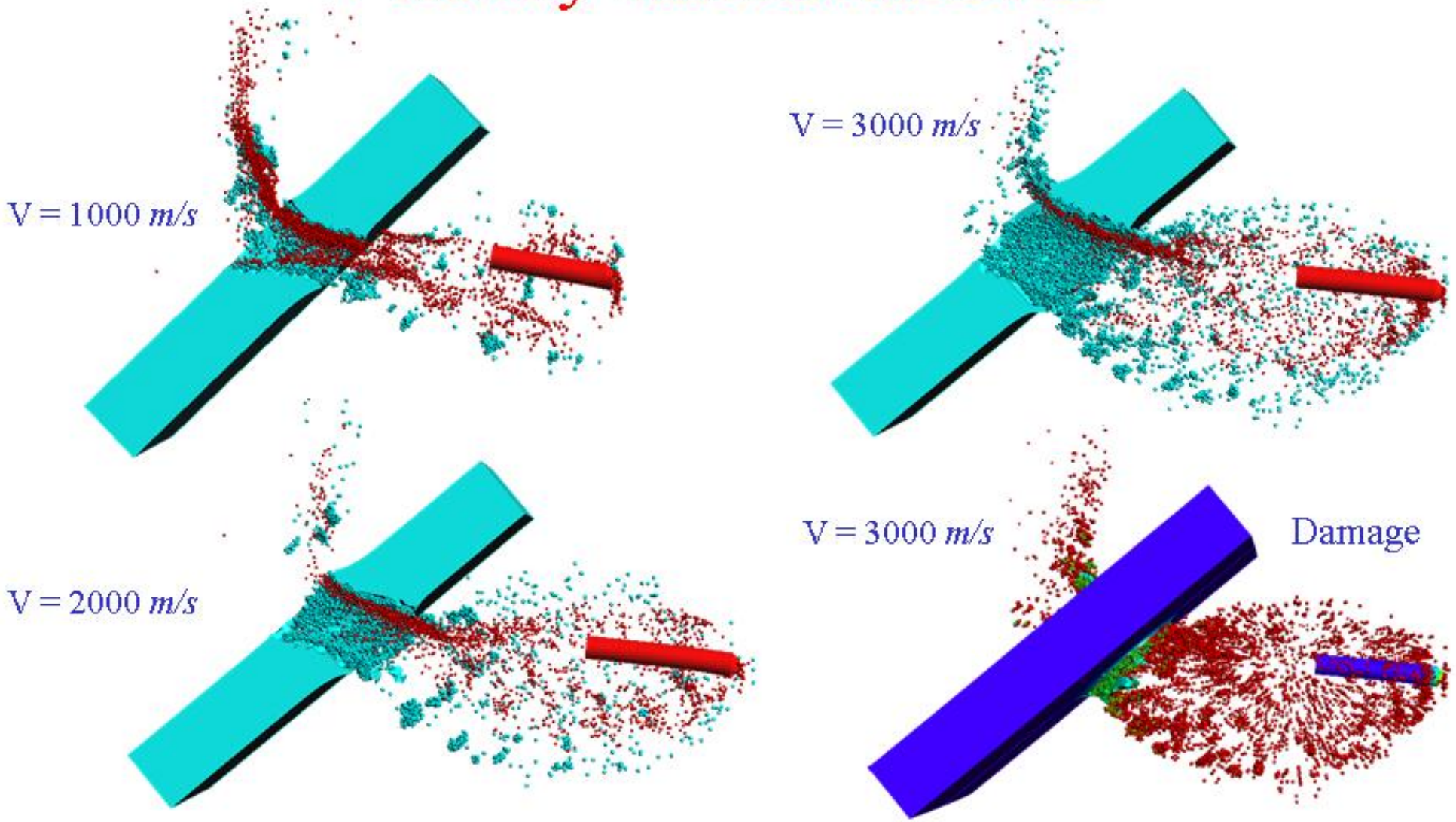


Damage



Rear view of aluminum witness plate

Velocity effects on BAD

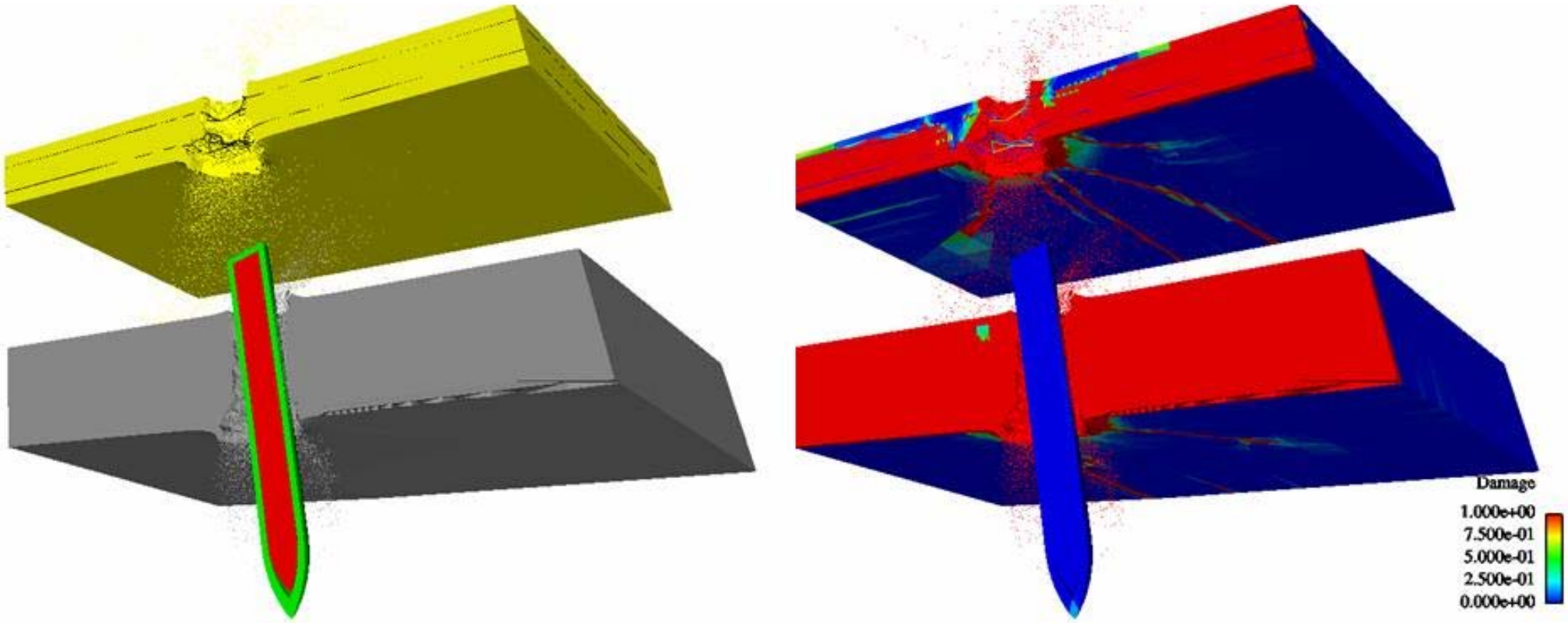


Projectile impacting concrete structure

t = 12 ms

Material

Damage



•12,413 particles generated (2.7 %, all concrete)

APS 2003

AHPCRC

NETWORK COMPUTING SERVICES, INC.

Projectile impacting SiC/Al layered target

At t = 0 μs:

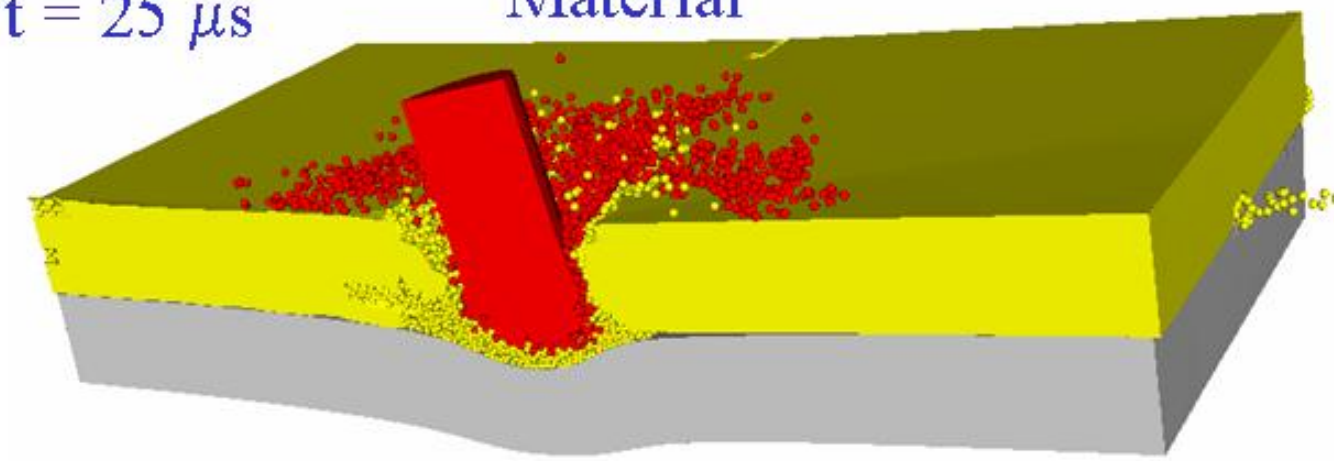
- Projectile
  - Hard steel cylinder
  - 2.4 cm length, 0.8 cm dia.
  - 850 m/s impact velocity
  - 8 degree obliquity
- Layered target
  - Top layer 0.7 cm SiC
  - Substrate 0.7 cm Al
- Initial mesh: 635,040 tetrahedra

At t = 25 μs:

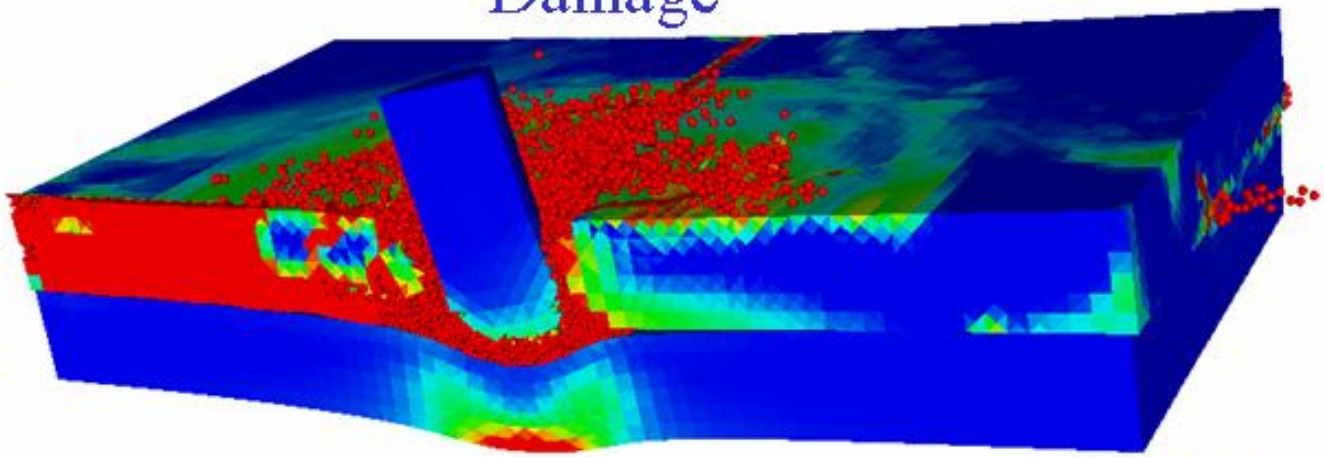
- Red particles on top due to dwell of projectile before penetration
- Yellow particles under projectile represent comminuted ceramic

t = 25 μs

Material



Damage



APS 2003

AHPCRC

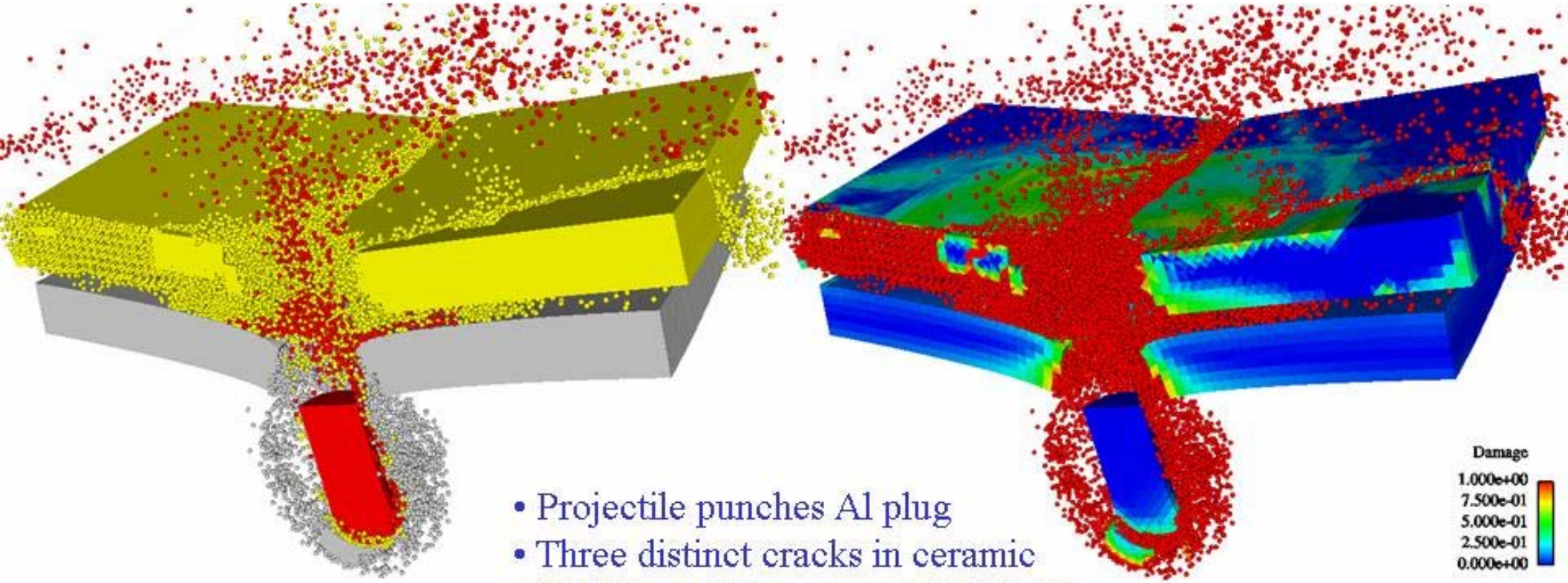
Network Computing Services, Inc.

Projectile impacting SiC/Al layered target

t = 100 μs

Material

Damage



- Projectile punches Al plug
- Three distinct cracks in ceramic
- 27,381 particles generated (4.3 %)

APS 2003

AHPCRC

Network Computing Services, Inc.

Summary and conclusions

- Robust, accurate and efficient computations can be performed for many ballistic impact conditions
- Key features in EPIC that provide this capability
  - Numerical algorithms for elements, particles, contact (Conversion of elements to particles is critical)
  - Material models (MMM subroutines)
  - Easy to use (Short Forms)
  - Visualization (Presto)
  - Parallelization (in progress)